OPIS PRZEDMIOTU ZAMÓWIENIA

I. Przedmiot zamówienia

Przedmiotem zamówienia jest zakup usługi dostępu do platformy kontenerowej Kubernetes do zarządzania klastrami wraz ze wsparciem technicznym na okres 12 miesięcy.

Przedmiot zamówienia dotyczy usług wymienionych w niniejszym OPZ, w ilościach zgodnych z poniższym wykazem lub równoważne zgodnie z warunkami określonymi w rozdziałach III-VI.

L.p.	Numer katalogowy	Produkt	Termin świadczenia usługi	Liczba produktów
1	MW01622	Red Hat OpenShift Platform Plus, Standard (2 Cores or 4 vCPU)	12 miesięcy	40
2	CU-GPS	Red Hat Consultant - Consulting Unit	12 miesięcy	200 roboczogodzin

II. Termin realizacji zamówienia

Rozpoczęcie świadczenia usług dostępu do platformy kontenerowej Kubernetes do zarządzania klastrami wraz ze wsparciem technicznym na okres 12 miesięcy nastąpi w terminie do 10 dni roboczych* **(***lub krótszym – w zależności od deklaracji przez Wykonawcę w formularzu ofertowym***) od dnia zawarcia umowy.**
W terminie, o którym mowa powyżej Wykonawca zapewni dostęp do spersonalizowanej strony pozwalającej upoważnionym ze strony Zamawiającego osobom na korzystanie ze wszystkich usług w ramach wykupionego dostępu do platformy kontenerowej Kubernetes do zarządzania klastrami wraz ze wsparciem technicznym na okres 12 miesięcy.

III. Warunki i forma płatności

Zamawiający wymaga możliwości rozliczania za **miesięczny** okres świadczenia dla niniejszej usługi dostępu do platformy kontenerowej Kubernetes do zarządzania klastrami wraz ze wsparciem technicznym na okres 12 miesięcy.

IV. Minimalne wymagania realizacji przedmiotu zamówienia

Zamawiający dopuszcza oferowanie produktów w pełni równoważnych do wymaganych.
W przypadku wystąpienia wątpliwości Zamawiającego, co do zaoferowanych produktów równoważnych, udowodnienie równoważności leży po stronie Wykonawcy. W przypadku zaoferowanie równoważnej usługi, narzędzia do zarządzania klastrami Kubernetesa w ramach usługi muszą spełniać opisane w niniejszym pkt III minimalne wymagania:

1. Minimalne wymagania funkcjonalne usługi dostępu do platformy kontenerowej:

1) Narzędzie musi mieć możliwość instalacji na platformach sprzętowych: x86, ARM, IBM Power, zgodnie z dostępną w internecie matrycą kompatybilności producenta platformy.
2) Narzędzie musi mieć możliwość instalacji w chmurze publicznej Amazon Web Service, Microsoft Azure, Google Cloud Engine, IBM Cloud i AlibabaCloud.
3) Narzędzie do instalacji platformy musi umożliwiać przeprowadzenie instalacji na wyżej wymienionych platformach w sposób zautomatyzowany tj. narzędzie do instalacji wygeneruje potrzebne do instalacji komponenty infrastruktury takie jak maszyny wirtualne i zainstaluje na nich platformę.
4) Narzędzie do instalacji platformy musi umożliwiać przeprowadzenie instalacji na wyżej wymienionych platformach w sposób manualny tj. w taki sposób że administrator może manualnie przygotować wszystkie komponenty infrastruktury potrzebne do instalacji platformy.
5) Narzędzie do instalacji musi umożliwiać instalację platformy na infrastrukturze bez dostępu do internetu oraz a infrastrukturze gdzie dostęp do internetu jest możliwy tylko przez serwery proxy.
6) Narzędzie musi umożliwiać instalację w konfiguracji wysokiej dostępności bez pojedynczego punktu awarii, gdzie każdy komponent platformy mający wpływ na jej dostępność będzie uruchomiony w co najmniej dwóch aktywnych instancjach.

7) Narzędzie musi umożliwiać instalację klastra rozciągniętego na więcej niż jeden niezależny ośrodek przetwarzania danych.
8) Narzędzie musi umożliwiać przeprowadzenie aktualizacji wersji oraz patchowanie platformy oraz systemu operacyjnego, na którym jest zainstalowana platforma w ramach jednolitej i automatycznej procedury aktualizacji.
9) W przypadku instalacji platformy na infrastrukturze bez dostępu do internetu istnieje możliwość przeprowadzenia jednolitej i automatycznej procedury aktualizacji platformy w oparciu o wcześniej pobraną aktualizację zgodnie z dokumentacją producenta platformy.
10) Narzędzie musi zawierać mechanizm skalowania węzłów klastra w sposób deklaratywny bez konieczności manualnej instalacji i konfiguracji węzłów.
11) Narzędzie musi umożliwiać izolację aplikacji przy użyciu technologii kontenerów w taki sposób, że na jednej instancji systemu operacyjnego równocześnie może być uruchomionych wiele odizolowanych aplikacji mających dostęp do ograniczonych zasobów systemowych takich jak pamięć RAM, moc procesora i system plików.
12) Do izolacji kontenerów na poziomie systemu operacyjnego Linux wykorzystywane są mechanizmy SElinux, Cgroups, Namespaces.
13) Narzędzie musi umożliwiać deklaratywne definiowanie limitów zasobów systemowych takich jak pamięć RAM, moc procesora i przepustowość sieci, które będą dostępne dla całej aplikacji jak i dla poszczególnych kontenerów aplikacji.
14) Narzędzie musi umożliwiać deklaratywne definiowanie globalnych limitów zasobów systemowych takich jak pamięć RAM, przestrzeń dyskowa i moc procesora, które są współdzielone przez wiele aplikacji.
15) Narzędzie musi zawierać wbudowany mechanizm umożliwiający automatyczną optymalizację konfiguracji limitów zasobów systemowych przypisanych do kontenerów w oparciu o analizę faktycznej konsumpcji zasobów przez te kontenery.
16) Narzędzie musi umożliwiać separację logiczną poszczególnych aplikacji, projektów (multitenancy) w taki sposób, że określony tenant może być odseparowany logicznie w warstwie sieciowej, systemu plików, węzłów klastra, dostępu do narzędzi administracyjnych i dla programistów oraz na poziomie systemu operacyjnego.
17) Narzędzie musi zawierać wbudowany rejestr obrazów OCI (Open ContainerInitiative).
18) Narzędzie musi pozwalać na uruchamianie aplikacji stanowych, które zapisują i odczytują dane z trwałego nośnika poprzez następujące interfejsy: NFS, Ceph RDB, CephFS, OpenstackCinder, iSCSi, Fibre Channel, Google GCE Volumes, Amazon EBS Volumes, Azure Disk, Azure File, VMWare VMDK.
19) Narzędzie musi pozwalać na wykorzystywanie przez aplikacji stanowe lokalnych zasobów dyskowych znajdujących się na węzłach klastra takich jak lokalne dyski, partycje i urządzenia blokowe.
20) Narzędzie musi umożliwiać instalację sterowników Kubernetes CSI (Container Storage Interface).
21) Narzędzie musi umożliwiać automatyczny dostęp aplikacjom kontenerowym do wyspecjalizowanych urządzeń i sterowników dostępnych na poszczególnych węzłach klastra takich jak np: GPU poprzez odpowiednie etykietowanie węzłów.
22) Narzędzie musi zawierać mechanizm tuningu węzłów klastra w celu optymalizowania ich wydajności pod kątem wymagań wydajnościowych uruchamianych aplikacji w oparciu o zdefiniowane profile konfiguracji węzłów klastra.
23) Narzędzie musi umożliwiać synchronizację czasu na węzłach klastra z wykorzystaniem protokołu NTP (Network Time Protocol) oraz przy użyciu zainstalowanych na węzłach klastra urządzeń PTP (Precision Time Protocol).
24) Narzędzie musi zawierać wbudowaną wewnętrzną wirtualną sieć (SDN) umożliwiającą komunikację pomiędzy aplikacjami i usługami uruchomionymi na platformie oraz dwukierunkową komunikację na zewnątrz.
25) Narzędzie musi umożliwiać konfigurację sieci wewnętrznej w taki sposób, żeby poszczególne aplikacje mogły być od siebie sieciowo odizolowane i jakakolwiek komunikacja pomiędzy aplikacjami była zablokowana.
26) Narzędzie musi umożliwiać w sieci wewnętrznej zastosowanie równocześnie adresacji IPv4 i IPv6.
27) Narzędzie musi umożliwiać mikro segmentację sieci wewnętrznej w taki sposób, że można precyzyjnie określić jakie usługi mogą się komunikować z innymi usługami z dokładnością do portu.

28) Narzędzie musi zawierać wbudowany moduł komunikacyjny (ingress router) umożliwiający komunikację protokołami HTTP, HTTPS, Websocket i TLS with SNI z aplikacjami uruchomionymi na platformie przez systemy uruchomione poza platformą oraz użytkowników aplikacji.
29) Narzędzie musi umożliwiać jednoczesne uruchomienie dwóch wersji aplikacji lub usługi i procentowe rozdzielenie ruchu sieciowego do poszczególnych wersji.
30) Narzędzie (ingress router) musi pozwalać na terminację SSL, reekrypcję SSL oraz przekazanie połączenia SSL bezpośrednio do kontenera.
31) Narzędzie musi umożliwiać uruchomienie dla aplikacji dedykowanego modułu komunikacyjnego (ingress router), który będzie obsługiwał tylko ruch przychodzący do danej aplikacji.
32) Narzędzie musi umożliwiać komunikację SSL w sieci wewnętrznej pomiędzy wybranymi usługami bez konieczności implementacji logiki komunikacji SSL w poszczególnych usługach.
33) Narzędzie musi umożliwiać szyfrowanie komunikacji w sieci wewnętrznej pomiędzy węzłami klastra przy użyciu IPSec lub TLS.
34) Narzędzie musi pozwalać na taką konfigurację aplikacji, żeby cały ruch sieciowy ze wszystkich usług danej aplikacji wychodził poza platformę tylko z jednego lub kilku dedykowanych dla danej aplikacji adresów IP bez względu na to, na którym węźle klastra dana usługa jest uruchomiona.
35) Narzędzie musi umożliwiać instalację certyfikowanych sterowników sieciowych Kubernetes CNI (Container Network Interface) pochodzących od różnych dostawców.
36) Narzędzie musi pozwalać na podpięcie wielu interfejsów sieciowych do jednego kontenera.
37) Narzędzie musi umożliwiać uruchamianie aplikacji dostarczanych w formie operatorów Kubernetes oraz Helmcharts.
38) Narzędzie musi umożliwiać budowanie kontenerów i uruchamianie aplikacji tworzonych w następujących technologiach: Node.js, Ruby, Perl, PHP, Python bez konieczności definiowania pliku Dockerfile.
39) Narzędzie musi umożliwiać budowanie i uruchamianie aplikacji tworzonych w technologii J2EE bez konieczności definiowania pliku Dockerfile.
40) Narzędzie musi umożliwiać budowanie kontenerów i uruchamianie aplikacji tworzonych w technologii Microsoft .NET Core bez konieczności definiowania pliku Dockerfile.
41) Narzędzie musi umożliwiać budowanie kontenerów i uruchamianie dowolnych bibliotek i platform programistycznych zgodnych z wyżej wymienionymi technologiami bez konieczności definiowania pliku Dockerfile.
42) Narzędzie musi zawierać wbudowane mechanizmy umożliwiające automatyzację budowania kontenerów, wdrożenia i uruchomienia aplikacji bezpośrednio z kodu źródłowego aplikacji bez konieczności definiowania plików Dockerfile.
43) Narzędzie musi zawierać gotowe szablony aplikacji, które umożliwiają po parametryzacji zbudowanie i uruchomienie na platformie aplikacji bez konieczności definiowania pliku Dockerfile oraz plików konfiguracyjnych Kubernetes.
44) Narzędzie musi zawierać gotowe narzędzia umożliwiające automatyczne zbudowanie aplikacji razem z zależnościami w formacie obrazu OCI.
45) Narzędzie musi umożliwiać skonteneryzowanie i uruchomienie aplikacji dostarczonych w postaci binarnej bez konieczności kompilacji kodu źródłowego i tworzenia pliku Dockerfile.
46) Obrazy kontenerów zbudowane na platformie muszą dawać możliwość uruchomienia zarówno na innych instancjach platformy jak i poza nią w dowolnym środowisku uruchomieniowym zgodnym z OCI.
47) Narzędzie musi zawierać gotowe narzędzia umożliwiające automatyczne zbudowanie obrazu kontenera opisanego plikiem konfiguracyjnym Dockerfile i jego uruchomienie na platformie.
48) Narzędzie musi zawierać i umożliwiać uruchomienie z gotowych obrazów OCI kontenera serwletówTomcat lub równoważnego zgodnego ze standardami javaservlet.
49) Narzędzie musi zawierać i umożliwiać uruchomienie z gotowych obrazów OCI aplikacji J2SE 1.8, 11 zbudowanych w oparciu o Spring Boot.
50) Narzędzie musi zawierać mechanizm optymalizacji działania aplikacji Java w kontenerach poprzez możliwość kompilacji aplikacji do wersji natywnej uruchamianej bezpośrednio jako proces w kontenerze bez konieczności uruchamiania maszyny wirtualnej Java (JVM) w tym kontenerze.

51) Narzędzie musi zawierać i umożliwiać uruchomienie z gotowych obrazów OCI serwer pojedynczego logowania (SSO) umożliwiający uwierzytelnianie i autoryzację przy użyciu protokołów OpenID Connect i SAML.
52) Narzędzie musi zawierać i umożliwiać uruchomienie z gotowych obrazów OCI baz danych MySQL, MariaDB, PostgreSQL, MongoDB.
53) Narzędzie musi zawierać wbudowane moduły do implementacji i automatyzacji procesu DevSecOps zgodnie z NIST SP 800-204C: https://csrc.nist.gov/publications/detail/sp/800-204c/final w szczególności:
    a) Narzędzie musi zawierać i umożliwiać uruchomienie na platformie serwera CI/CD Tekton, Jenkins;
    b) Narzędzie musi zawierać i umożliwiać uruchomienie na platformie serwera GitOpsArgoCD;
    c) Wbudowany rejestr obrazów kontenerów zgodnych z OCI oraz bibliotekę wspieranych przez dostawcę obrazów bazowych OCI;
    d) Moduł bezpieczeństwa umożliwiający skanowanie obrazów i bibliotek programistycznych pod kątem występowania luk bezpieczeństwa, analizowanie plików konfiguracyjnych Kubernetes pod kątem bezpieczeństwa, monitorowanie działających w kontenerach procesów i ruchu sieciowego w kontenerach;
    e) Silnik polityk umożliwiający implementację i egzekwowanie polityk bezpieczeństwa w procesie DevSecOps;
    f) Wbudowany dedykowany moduł do zarządzania i monitorowania komunikacji sieciowej dla aplikacji zbudowanych w architekturze mikro usług (Service Mesh);
    g) Moduł do ciągłego monitorowania zgodności konfiguracji klastra z CIS Kubernetes lub równoważnym;
    h) Moduł do monitoring aplikacji i śledzenia ruchu wewnątrz aplikacji;
    i) Moduł do agregacji logów aplikacji i platformy;
    j) Wszystkie wyżej wymienione komponenty muszą być konfigurowane jako kod i zautomatyzowane przy użyciu dedykowanych narzędzi do automatyzacji.
54) Narzędzie musi zawierać katalog aplikacji umożliwiający uruchomienie umieszczonych tam aplikacji bazujących na operatorach Kubernetes, Helmcharts oraz innych mechanizmach umożliwiających tworzenie szablonów aplikacji.
55) Dostawca narzędzia dostarcza rejestr certyfikowanych operatorów Kubernetes, które mogą być instalowane na platformie manualnie przy użyciu narzędzi administracyjnych wchodzących w skład platformy oraz automatycznie przy użyciu dedykowanych narzędzi do automatyzacji.
56) Narzędzie musi dawać dostęp do publicznego rejestru obrazów, z którego można pobrać stale aktualizowane i certyfikowane przez dostawcę platformy wyżej wymienione obrazy OCI
57) Narzędzie musi zawierać i umożliwiać uruchomienie centralnego serwera agregacji logów aplikacji i platformy opartego na technologii ElasticSearch, Kibana i Fluentd lub równoważnych który umożliwia długotrwałe przechowywanie logów na trwałych nośnikach danych.
58) Narzędzie musi posiadać wbudowany mechanizm umożliwiający przesyłanie logów do zewnętrznych systemów agregacji i analizy logów takich jak ElasticSearch, Fluentd, Syslog, Kafka, Loki, AWS Cloudwatch.
59) Narzędzie musi zawierać i umożliwiać uruchomienie centralnego serwera agregacji metryk aplikacji działających na platformie oraz samej platformy opartego na technologii Prometheus lub równoważnej, który umożliwia długotrwałe przechowywanie metryk na trwałych nośnikach danych.
60) Narzędzie powinna umożliwiać zbieranie i przechowywanie metryk oraz logów aplikacji przez określony czas.
61) W przypadku uruchamiania aplikacji z obrazów OCI muszą one pozwalać na uruchomienia jako użytkownik systemowy bez pełnych praw administracyjnych.
62) Narzędzie musi domyślnie uniemożliwić uruchomienie kontenerów na prawach użytkownika root.
63) Narzędzie musi pozwalać na zautomatyzowane przenoszenie aplikacji pomiędzy różnymi instancjami platformy, które mogą być uruchomione na różnych infrastrukturach (serwery fizyczne, wirtualne, chmura prywatna, publiczna).

64) Narzędzie musi umożliwiać uruchomienie nowej wersji aplikacji przy zachowaniu pełnej dostępności aplikacji i bez konieczności jej zatrzymania lub ograniczenia dostępności (rolling upgrade).
65) Narzędzie musi umożliwiać automatyczne cofnięcie wdrożenia aplikacji (deployment) do jednej z poprzednich wersji.
66) W przypadku klastrowania aplikacji narzędzie musi zapewniać mechanizm rozłożenia ruchu pomiędzy instancjami aplikacji (loadbalancing).
67) Narzędzie musi umożliwiać podłączenie zewnętrznych komponentów do rozkładania ruchu pomiędzy instancjami aplikacji (zewnętrzny loadbalancer).
68) Narzędzie musi umożliwiać uruchamiania wielu aplikacji równocześnie na współdzielonych zasobach sprzętowych.
69) Narzędzie musi zawierać wbudowany mechanizm skalowania, który pozwala określić deklaratywnie ile instancji danej aplikacji ma być uruchomionych jednocześnie i pozwala na skalowanie ilości uruchomionych jednocześnie instancji aplikacji.
70) Narzędzie musi zawierać wbudowany mechanizm do wdrażania aplikacji w którym skalowanie aplikacji odbywa się dynamicznie w sposób zautomatyzowany bazując na ruchu generowanym do danej aplikacji lub wydajności instancji aplikacji.
71) Narzędzie musi zawierać wbudowany mechanizm obsługi zdarzeń umożliwiający automatyczne skalowanie aplikacji w odpowiedzi na pojawiające się zdarzenia, których źródłem mogą być systemy messagingnp: Kafka lub http.
72) Narzędzie musi zawierać wbudowane mechanizmy automatycznego skalowania aplikacji (uruchamiania lub wyłączania kolejnych instancji aplikacji) w oparciu o metryki zużycia zasobów systemowych przez aplikację.
73) Narzędzie musi zawierać wbudowaną konsolę administracyjną umożliwiającą wykonywanie zadań administracyjnych przez przeglądarkę internetową.
74) Narzędzie musi zawierać wbudowane narzędzia umożliwiające administrację i konfigurację platformy z poziomu linii poleceń działające na Microsoft Windows, Red Hat Enterprise Linux, MacOSX.
75) Narzędzie musi zawierać wbudowany webowy terminal znakowy, który umożliwia dostęp do narzędzia do administracji i konfiguracji platformy z poziomu linii poleceń poprzez przeglądarkę internetową.
76) Narzędzie musi zawierać wbudowany interfejs programistyczny API dostępny przez protokół REST umożliwiający administrację platformą przy użyciu narzędzi zewnętrznych.
77) Narzędzie musi zawierać wbudowane mechanizmy uwierzytelniania i autoryzacji użytkowników oparte na OAuth 2.0, oraz umożliwia konfigurację dostępu opartego na rolach dla różnych grup użytkowników w tym administratorów i programistów.
78) Narzędzie musi umożliwiać definiowanie różnych projektów dla poszczególnych aplikacji i przypisywania uprawnień do nich dla określonych grup użytkowników.
79) Narzędzie musi pozwalać na integrację z zewnętrznymi bazami użytkowników w tym Microsoft Active Directory lub LDAP oraz serwerami autoryzacji zgodnymi z OAuth 2.0.
80) Narzędzie musi zawierać wbudowany mechanizm umożliwiający administratorom określenie uprawnień dla uruchamianych na platformie kontenerów takich jak uprawnienia użytkownika, dostępu do zasobów sprzętowych oraz profile seccomp.
81) Narzędzie musi umożliwiać przechowywanie konfiguracji klastra i aplikacji Kubernetes na trwałych nośnikach danych w formie zaszyfrowanej.
82) Narzędzie musi zawierać mechanizm konfiguracji systemu operacyjnego z poziomu platformy bez konieczności manualnej konfiguracji bezpośrednio na systemie operacyjnym.
83) Narzędzie musi zawierać elastyczny silnik polityk, który umożliwia definiowanie i egzekwowanie polityk konfiguracji platformy i aplikacji wdrożonych na platformie.
84) Narzędzie musi zawierać wbudowany mechanizm proaktywnego wykrywania, priorytetowania i rozwiązywania problemów wydajnościowych, stabilności i bezpieczeństwa platformy.
85) Narzędzie musi zawierać wbudowany dedykowany moduł do zarządzania i monitorowania komunikacji sieciowej dla aplikacji zbudowanych w architekturze mikro usług.
86) Narzędzie do zarządzania komunikacją sieciową musi umożliwiać zarządzania ruchem wchodzącym i wychodzącym, uwierzytelnianie, autoryzację i szyfrowanie ruchu przez mTLS, możliwość filtrowania ruchu i zarządzania nim w oparciu o zdefiniowane przez administratora reguły.

87) Narzędzie do zarządzania komunikacją sieciową musi posiadać wbudowaną konsolę webową umożliwiającą konfigurację i wizualizację komunikacji wewnątrz Service Mesh.
88) Narzędzie do zarządzania komunikacją sieciową musi umożliwiać uruchomienie na jednym klastrze wielu niezależnych instancji sieci wraz z oddzielnymi konsolami do zarządzania dla każdej instancji.
89) Narzędzie musi zawierać wbudowany mechanizm śledzenia komunikacji pomiędzy usługami uruchomionymi na platformie zgodnej z OpenTracing API.
90) Narzędzie musi zawierać zintegrowane środowisko programistyczne (IDE), które umożliwia rozwijanie kodu aplikacji, jego kompilację i uruchomienie na platformie bez konieczności wcześniejszej jej konteneryzacji.
91) Narzędzie musi posiadać moduł umożliwiający balansowanie obciążenia poszczególnych węzłów klastra w celu optymalizacji konsumpcji zasobów.
92) Narzędzie musi posiadać narzędzie umożliwiające migrację aplikacji (konfiguracji i danych) pomiędzy różnymi klastrami.
93) Wszystkie oferowane komponenty narzędzia muszą być oferowane w ramach jednolitego rozwiązania oraz są objęte wsparciem producenta i nie będą dodatkowo instalowane przez dostawcę w ramach wdrożenia będącego przedmiotem tego postępowania. Wszystkie platformy kontenerowe traktowane są jako środowiska produkcyjne.

2. Wymagania funkcjonalne narzędzia do przechowywania obrazów kontenerów:

1) Narzędzie do obrazów musi mieć możliwość zainstalowania na klastrze Kubernetes w formie skonteneryzowanej.
2) Narzędzie do obrazów musi mieć możliwość zainstalowania systemie operacyjnym Linux na maszynach wirtualnych lub serwerach fizycznych.
3) Narzędzie do obrazów musi umożliwiać skanowanie zawartości obrazów OCI pod kątem występowania luk bezpieczeństwa.
4) Narzędzie do obrazów musi umożliwiać ciągłe automatyczne skanowanie obrazów w określonych interwałach czasowych w celu ciągłego wykrywania luk bezpieczeństwa.
5) Narzędzie do obrazów musi umożliwiać prezentację wyników skanowania obrazów OCI bezpośrednio w interfejsie użytkownika klastra Kubernetes.
6) Narzędzie do obrazów musi umożliwiać automatyczne wysyłanie powiadomień w przypadku wykrycie nowej podatności o określonym poziomie (severity) w obrazie, przesłania obrazu do repozytorium lub uruchomienia procedury budowania obrazu.
7) Narzędzie do obrazów musi pozwalać na replikację repozytoriów pomiędzy różnymi instancjami Rejestru rozproszonymi geograficznie.
8) Narzędzie do obrazów musi pozwalać na przechowywanie obrazów na różnych typach przestrzeni dyskowej zarówno lokalnych jak i w chmurach publicznych.
9) Narzędzie do obrazów musi zawierać log audytowy, który umożliwia śledzenie zdarzeń i akcji wywołanych zarówno przez API jaki i interfejs użytkownika.
10) Narzędzie do obrazów musi umożliwiać uruchomienie w konfiguracji wysokiej dostępności bez pojedynczego punktu awarii.
11) Narzędzie do obrazów musi posiadać wbudowany mechanizm uwierzytelniania, który umożliwia uwierzytelnianie użytkowników LDAP oraz przy użyciu protokołów OAuth 2.0 i OpenID Connect.
12) Narzędzie do obrazów musi pozwalać na przypisanie ról i uprawnień użytkownikom z rozróżnieniem na administratorów platformy oraz administratorów i użytkowników poszczególnych repozytoriów.
13) Narzędzie do obrazów musi umożliwiać monitorowanie i dostęp do metryk poprzez bazę metryk Prometheus.
14) Narzędzie musi pozwalać na przesyłanie powiadomień o wystąpieniu różnych zdarzeń na platformie oraz w poszczególnych repozytoriach przez Email, Slack oraz Webhook.
15) Narzędzie musi pozwalać na utworzenie kont serwisowych, którym mogą być przypisane różne uprawniania na poziomie platformy oraz poszczególnych repozytoriów w celu używania przez zewnętrzne aplikacje.
16) Narzędzie musi umożliwiać jednoczesne przechowywanie obrazów dla różnych typów architektur sprzętowych takich jak x86, IBM Power LE oraz Z System, ARM, Windows.

3. Wymagania funkcjonalne narzędzia do zabezpieczania kontenerów:

1) Narzędzie musi mieć możliwość zainstalowania w klastrze Kubernetes w formie skonteneryzowanej i monitorować wiele klastrów zdalnych przy użyciu agentów zainstalowanych lokalnie na tych klastrach.
2) Narzędzie musi udostępniać interfejs użytkownika przez przeglądarkę internetową oraz interfejs programistyczny API.
3) Narzędzie musi umożliwiać śledzenie i wizualizację ruchu sieciowego wewnątrz klastra oraz połączeń na zewnątrz klastra z możliwością filtrowania ruchu do poziomu projektów (namespaces), wdrożeń (deployments) i poszczególnych podów.
4) Narzędzie musi umożliwiać śledzenie procesów uruchomionych w kontenerach i wykrywanie aktywności niezgodnych ze zdefiniowanymi politykami bezpieczeństwa.
5) Narzędzie musi umożliwiać ciągłe skanowanie obrazów w celu wykrycia znanych podatności w bibliotekach systemowych oraz aplikacyjnych uruchamiane automatycznie w określonych interwałach czasowych.
6) Narzędzie musi umożliwiać egzekwowanie zgodności z politykami bezpieczeństwa na każdym etapie życie aplikacji: podczas budowania obrazów kontenerów, podczas wdrażania aplikacji na klastrze i w trakcie działania aplikacji.
7) Narzędzie musi pozwalać na weryfikację zgodności ze standardami i regulacjami takimi jak CIS Benchmarks, Payment Card Industry (PCI), HealthInsurancePortability and AccountabilityAct (HIPAA) oraz NIST SP 800-190.
8) Narzędzie musi udostępniać raporty zgodności z wyżej wymienionymi standardami i regulacjami oraz umożliwia eksport tych raportów do oddzielnych plików w celu udostępnienia audytorom.
9) Narzędzie musi umożliwiać filtrowanie informacji o zgodności z poszczególnymi standardami i regulacjami na poziomie klastra, węzłów klastra lub projektów (namespaces).
10) Narzędzie musi umożliwiać automatyczne generowanie polityk sieciowych na podstawie śledzenia ruchu i polityk bezpieczeństwa.
11) Narzędzie musi umożliwiać symulację polityk sieciowych przed ich wdrożeniem w celu analizy ich wpływu na działające aplikacje.
12) Narzędzie musi pozwalać na priorytetyzację ryzyka działania poszczególnych aplikacji na platformie.
13) Narzędzie musi dostarczać gotowe polityki bezpieczeństwa w celu automatycznego wykrywania niezgodności w konfiguracji sieciowej, eskalacji uprawnień w kontenerach, wykrywania procesów uruchamianych jako root i podobnych.
14) Narzędzie musi umożliwiać analizę uprawnień Kubernetes role-basedaccesscontrol (RBAC) przypisanych do użytkowników i kont serwisowych (service accounts).
15) Narzędzie musi umożliwiać śledzenie zdarzeń w klastrze zapisywanych w Kubernetes audit log w celu wykrywania niezgodności z politykami bezpieczeństwa.
16) Narzędzie musi dostarczać gotowe polityki bezpieczeństwa w celu automatycznego wykrywania oprogramowania cryptomining, eskalacji uprawnień i znanych podatności.
17) Narzędzie musi udostępniać API oraz umożliwiać integrację z zewnętrznymi systemami CI/CD i DevOps, skanerami obrazów, rejestrami obrazów, systemami SIEM i systemami do powiadamiania.
18) Narzędzie musi posiadać mechanizmy do zablokowanie uruchomienia kontenera z obrazu, którego zawartość jest nieznana lub zawiera lukę bezpieczeństwa.
19) Narzędzie musi obsługiwać wszystkie platformy kontenerowe w środowisku Zamawiającego.

4. Wymagania funkcjonalne narzędzia do zarządzania środowiskiem hybrydowym (wiele klastrów na wielu infrastrukturach)

1) Narzędzie musi mieć możliwość zainstalowania w klastrze Kubernetes w formie skonteneryzowanej.
2) Narzędzie musi udostępniać interfejs użytkownika przez przeglądarkę internetową oraz interfejs programistyczny API.
3) Narzędzie musi umożliwiać instalację nowych klastrów Kubernetes oraz zarządzanie istniejącymi klastrami Kubernetes zarówno w infrastrukturze zamawiającego jak i w chmurze publicznej.
4) Narzędzie musi umożliwiać instalację nowych klastrów Kubernetes na platformie AWS, Microsoft Azure, Google Cloud Platform, Microsoft AzureGovernment, serwery fizycznie, Red HatOpenStack Platform, VMwarevSphere.

5) Narzędzie musi pozwalać na skalowanie węzłów na zarządzanych klastrach.
6) Narzędzie musi pozwalać na definiowanie uprawnień do zarządzania oddzielnie różnymi klastrami i grupami klastrów.
7) Narzędzie musi umożliwiać zbieranie, długoterminowe przechowywanie, retencję i wizualizację metryk wydajności działania zarządzanych klastrów.
8) Narzędzie musi umożliwiać definiowanie własnych metryk i ich wizualizację na platformie oraz na definiowanie własnych wykresów na platformie.
9) Narzędzie musi umożliwiać wyszukiwanie obiektów Kubernetes wchodzących w skład danego klastra lub aplikacji.
10) Narzędzie musi zawierać mechanizm alertowania i wysyłania powiadomień w przypadku wygenerowania alertów.
11) Narzędzie musi zawierać gotowe polityki do konfiguracji zarządzanych klastrów oraz umożliwiające automatyczną instalację dodatkowych komponentów na zarządzanych klastrach przy użyciu operatorów Kubernetes lub Helmcharts.
12) Narzędzie musi umożliwiać centralne wdrażanie poprawek dla wykrytych niezgodności z politykami i standardami na zarządzanych klastrach.
13) Narzędzie musi umożliwiać monitorowanie prawidłowego działanie aplikacji zainstalowanych na zarządzanych klastrach oraz wgląd w szczegóły konfiguracji aplikacji.
14) Narzędzie musi pozwalać na automatyczne wdrażanie aplikacji na zarządzanych klastrach zgodnie ze zdefiniowanymi regułami przyporządkowania aplikacji do klastrów.
15) Narzędzie musi pozwalać na automatyczne wdrażanie aplikacji Helm na zarządzanych klastrach zgodnie ze zdefiniowanymi regułami przyporządkowania aplikacji do klastrów.
16) Narzędzie musi udostępniać graficzne narzędzie dostępne przez przeglądarkę internetową umożliwiające konfigurację procedury wdrażania aplikacji na zarządzanych klastrach umożliwiające wybór źródła zawierającego konfigurację aplikacji oraz reguł wyboru klastrów na które aplikacja ma być wdrożona.
17) Narzędzie musi umożliwiać wdrażanie aplikacji na zarządzanych klastrach z wykorzystaniem zewnętrznego narzędzia GitOpsnp: ArgoCD lub równoważnego.
18) Narzędzie musi pozwalać na wykonanie kopii zapasowej konfiguracji klastra i jego odzyskanie na innym klastrze.
19) Narzędzie musi obsługiwać wszystkie platformy kontenerowe w środowisku Zamawiającego.

5. Wymagania funkcjonalne narzędzia integracji platformy kontenerowej z rozwiązaniami obsługi zasobów dyskowych

1) Narzędzie musi mieć możliwość tworzenia wolumenów Kubernetes (Persistent Volume) z trybem dostępu RWO i RWX.
2) Narzędzie musi umożliwiać tworzenie wolumenów Kubernetes typu blokowego i plikowego.
3) Narzędzie musi umożliwiać tworzenie woluminów obiektowych (Object Bucket).
4) Utworzone wolumeny obiektowe muszą być obsługiwane przez interfejs S3.
5) Wolumeny obiektowe muszą pozwalać na replikowanie do innych obiektowych systemów pamięci masowych zarówno lokalnych jak i w chmurze publicznej.
6) Narzędzie musi umożliwiać zarządzanie procesem dostarczanie danych rozwiązania dyskowego zarządzanego programowo dla całej platformy kontenerowej.

6. Wymagania funkcjonalne narzędzia do udostępniania warstwy przechowywania

1) Narzędzie musi posiadać możliwość serwowania danych za pomocą interfejsów blokowych, plikowych i obiektowych.
2) Narzędzie musi działać w formie klastera i gwarantować wydajność, niezawodność i skalowalność.
3) Utworzone wolumeny obiektowe muszą być obsługiwane przez interfejs S3.
4) Narzędzie musi umożliwiać asynchroniczną replikację danych pomiędzy różnymi klastrami w tym Kubernetes.
5) Narzędzie musi zawierać mechanizm umożliwiający kontrolę i zarządzanie pojemnością danych w rozwiązaniu.
6) Narzędzie musi zapewniać niezawodność przechowywanych danych poprzez posiadanie dwóch lub więcej replik danych oraz erasurecoding.

7) Narzędzie musi zapewniać odporność na awarie i minimalizację prac wymaganych przy utrzymaniu rozwiązania
8) Narzędzie musi pozwalać na elastyczną zmianę polityk ochrony danych w trakcie pracy.
9) Narzędzie musi umożliwiać zbudowanie współdzielonej przestrzeni dyskowej w oparciu o dyski wewnętrzne serwerów fizycznych. Narzędzie powinno wspierać następujące konfiguracje serwerów: hybrydowa w oparciu o dyski SSD i HDD oraz all-flash w oparciu o dyski SSD (SAS/SATA/NVMe).
10) Klaster, na którym zostanie zainstalowane Rozwiązanie, musi umożliwiać utworzenie przestrzeni dyskowej złożonej z co najmniej 16 hostów i rozbudowę każdego z hostów do co najmniej 30 dysków
11) Narzędzie musi umożliwiać konfigurację serwerów all-NVMe.
12) Narzędzie musi umożliwiać zmniejszenie lub zwiększenie przestrzeni dyskowej poprzez: usunięcie, lub dodanie pojedynczego dysku, dwóch i więcej dysków, usunięcie lub dodanie serwera fizycznego w sposób niewymagający przestoju i przerwy w działaniu.
13) Narzędzie musi zapewniać możliwość obsługiwania woluminów blokowych do rozmiaru co najmniej 100TB.
14) Narzędzie musi zapewniać funkcjonalność konfigurowalnych mechanizmów zabezpieczania danych na wypadek awarii sprzętowej pojedynczego dysku, węzła, szafy rack oraz całego data center.
15) Lista wspieranych i certyfikowanych konfiguracji serwerów kompatybilnych z rozwiązaniem musi być zamieszczona na oficjalnej stronie producenta narzędzia. Wymagane jest wsparcie dla min. trzech niezależnych producentów sprzętu serwerowego dostępnego na terenie Unii Europejskiej.
16) Narzędzie musi działać w środowiskach bez dostępu do sieci Internet.
17) Narzędzie musi zapewniać możliwość zarządzania użytkownikami i rolami użytkowników (RBAC).
18) Narzędzie musi umożliwiać udostępnianie przestrzeni dyskowej również dla fizycznych serwerów, w oparciu o technologię iSCSI, a także umożliwiać zarządzanie dostępnością, pojemnością i wydajnością bez konieczności ograniczania dostępu do danych.
19) Narzędzie musi zawierać interfejs API umożliwiający automatyzowanie wdrażania lub modyfikacji konfiguracji Systemu.
20) Narzędzie musi być wspierane jako backup target co najmniej dla 2 producentów systemu backup: CommVault, IBM Spectrum Protect Plus, IBM Spectrum Protectserver, NetAppAltaVault, RubrikCloud Data Management (CDM), Trilio, Veeam (objectstorage), Veritas NetBackup for Symantec OpenStorage (OST) cloud backup.
21) Narzędzie musi umożliwiać tworzenie woluminów Kubernetes (Persistent Volume) z trybem dostępu RWO i RWX.
22) Narzędzie musi umożliwiać tworzenia woluminów Kubernetes typu blokowego i plikowego.
23) Narzędzie musi zawierać wbudowany mechanizm kompresji danych.
24) Narzędzie musi umożliwiać klonowanie i tworzenie migawek (snapshot) woluminów danych.
25) Narzędzie musi umożliwiać zdalną replikację danych typu on-line (bez przerywania prezentacji zasobów dyskowych) do rozwiązania tej samej rodziny w trybie asynchronicznym.
26) Narzędzie musi zapewniać szyfrowanie danych na poziomie całego klastra.
27) Narzędzie musi umożliwiać uruchomienie warstwy danych w konfiguracji HA z 3 replikami danych.
28) Narzędzie musi zapewniać mechanizm bezpiecznego, bezprzerwowego i automatycznego uaktualniania rozwiązania wliczając w to zarówno poprawki bezpieczeństwa, jak i zmianę wersji oprogramowania.
29) Narzędzie musi posiadać narzędzia kryptograficzne zgodne ze standardem FIPS 140-2.
30) System operacyjny rozwiązania musi posiadać udokumentowany certyfikat bezpieczeństwa zgodny ze standardem ISO/IEC 15408.

7. Minimalne wymagania w zakresie wsparcia technicznego

W ramach przedmiotu zamówienia Wykonawca zapewni wsparcie techniczne na okres 12 miesięcy, świadczone według poniżej opisanych minimalnych wymaganiach:

1) analiza, wykonanie i wsparcie w konfiguracji elementów platformy RedHat OpenShift Platform Plus;
2) analiza, wykonanie i wsparcie w konfiguracji procesów CI/CD w ramach platformy RedHat Openshift Platform Plus;
3) wsparcie w analizie i konfiguracji elementów platformy odpowiedzialnych za warstwy bezpieczeństwa, sieci i aplikacji;
4) wsparcie w analizie i konfiguracji rozwiązań zintegrowanych systemu odpowiedzialnych za procesy CI/CD, katalug usług, podsystemy metryk, agregacji logów, rozproszonego storage;
5) wsparcie w analizie i konfiguracji rozwiązań RBAC (Role-based Access Controll), separacji przestrzeni nazw;
6) wsparcie w analizie i konfiguracji bezprzerwowych procesów aktualizacji minimalizując niedostępność zasobów oraz minimalizację samego procesu upgrade-u.
7) dostęp do bazy wiedzy oraz pobierania aktualizacji znajdujących się na stronie producenta
8) Obsługi nieograniczonej liczby zgłoszeń błędów, wad, awarii oraz konsultacji w zakresie konfiguracji, funkcjonalności, obsługi systemu pocztowego świadczone w dni robocze w godz. 8.00.16.00 w siedzibie Zamawiającego lub zdalnie poprzez bezpieczne szyfrowane połączenie.
9) Zgłoszeniem w ramach wsparcia technicznego jest każde zgłoszenie błędów, wad, awarii oraz konsultacji w zakresie platformy skierowane przez Zamawiającego do Wykonawcy poprzez jeden z dostępnych kanałów komunikacji, wymienionych w pkt 10).
10) Zgłoszenia przyjmowane będą przez Wykonawcę pod podany w umowie:
    a) nr telefonu,
    b) adresem e-mail,
    c) serwisem www.
11) Wykonawca niezwłocznie po otrzymaniu zgłoszenia przystąpi do jego rozpoznania i podejmie działania zmierzające do rozwiązania.
12) Fakt przystąpienia do rozpoznania zgłoszenia będzie potwierdzany przez Wykonawcę drogą elektroniczną na adres wskazany przez Zamawiającego niezwłocznie od chwili przekazania zgłoszenia przez Zamawiającego jednak nie później niż do 60 minut.
13) Wykonawca zobowiązuje się zapewnić rozwiązanie (lub w przypadku błędów oprogramowania eskalowanie do producenta) zgłoszonej awarii najpóźniej do końca następnego dnia roboczego od momentu zgłoszenia przez Zamawiającego, przy czym przez rozwiązanie rozumie się całkowite usuniecie awarii.
14) Wykonawca zapewni nadzorowanie zgłoszeń eskalowanych do producenta.

8. Minimalne wymagania w zakresie dodatkowych usług konsultacyjnych:

1) Wykonawca zapewni dodatkowe usługi konsultacyjne świadczone w wymiarze maksimum 200 roboczogodzin przez zespół składający się z certyfikowanych przez Red Hat specjalistów, w którym co najmniej jeden posiada certyfikat Red Hat Certified Specialist in Security Containers and Openshift Conainer Platform lub równoważny.
2) Zespół musi posługiwać się płynnie językiem polskim.