Szczegółowy opis przedmiotu zamówienia

I. Przedmiot zamówienia

Przedmiotem zamówienia jest zakup subskrypcji oprogramowania Red Hat OpenShift Platform Plus (MW01622), wraz ze wsparciem producenta na poziomie Standard – 14 szt., obowiązujących przez okres 36 miesięcy.

Zamawiający dopuszcza dostarczenie oprogramowania równoważnego spełniającego wymagania określone w pkt II SOPZ.

II. Opis wymagań dla oprogramowania równoważnego.

1. Narzędzie musi umożliwiać instalację w środowiskach zwirtualizowanych platformy VMware vSphere 8 zamawiającego opartych o architekturę sprzętową: x86.
2. Narzędzie do instalacji platformy musi umożliwiać przeprowadzenie instalacji w sposób zautomatyzowany, tj. narzędzie do instalacji wygeneruje potrzebne do instalacji komponenty infrastruktury takie jak maszyny wirtualne i zainstaluje na nich platformę.
3. Narzędzie do instalacji platformy musi umożliwiać przeprowadzenie instalacji, w taki sposób, że administrator może manualnie przygotować wszystkie komponenty infrastruktury potrzebne do instalacji platformy.
4. 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.
5. Narzędzie musi umożliwiać przeprowadzenie aktualizacji wersji oraz patchowanie platformy oraz systemu operacyjnego, na którym jest zainstalowana platforma w ramach jednolitej i automatycznej procedury aktualizacji.
6. Narzędzie musi zawierać mechanizm skalowania węzłów klastra w sposób deklaratywny bez konieczności manualnej instalacji i konfiguracji węzłów.
7. Narzędzie musi zawierać mechanizm konfiguracji systemu operacyjnego z poziomu platformy bez konieczności manualnej konfiguracji bezpośrednio na systemie operacyjnym.
8. 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.
9. Narzędzie musi zawierać wbudowany mechanizm proaktywnego wykrywania, priorytetowania i rozwiązywania problemów wydajnościowych, stabilności i bezpieczeństwa platformy.

10. 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).
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. System operacyjny węzłów klastra musi być oparty na jądrze typu Linux/Unix.
13. Do izolacji kontenerów na poziomie systemu operacyjnego Linux wykorzystywane są mechanizmy SElinux, Cgroups, Namespaces.
14. 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.
15. Narzędzie musi umożliwiać uruchamianie aplikacji dostarczanych w formie manifestów Kubernetes, operatorów Kubernetes oraz Helmcharts.
16. Narzędzie musi umożliwiać uruchomienie gotowych obrazów kontenerów także z niezależnych od platformy repozytoriów obrazów zgodnych z OCI (Open Container Initiative).
17. Narzędzie musi pozwalać na zautomatyzowane przenoszenie aplikacji pomiędzy różnymi instancjami platformy.
18. Narzędzie musi umożliwiać uruchomienie nowej wersji aplikacji przy zachowaniu pełnej dostępności aplikacji i bez konieczności jej zatrzymania lub ograniczenia dostępności (rolling upgrade).
19. Narzędzie musi umożliwiać automatyczne cofnięcie wdrożenia aplikacji (deployment) do jednej z poprzednich wersji.
20. W przypadku klastrowania aplikacji narzędzie musi zapewniać mechanizm rozłożenia ruchu pomiędzy instancjami aplikacji (loadbalancing).
21. Narzędzie nie może wykluczać wykorzystania zewnętrznych komponentów do rozkładania ruchu w celu dostępu do aplikacji uruchomionych wewnątrz klastrów (zewnętrzny loadbalancer).
22. 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.

23. 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.
24. Narzędzie musi zawierać wbudowane mechanizmy automatycznego skalowania aplikacji (uruchamiania lub wyłączania kolejnych instancji aplikacji) w oparciu o metryki zużycia zasobów systemowych przez aplikację.
25. Narzędzie musi zawierać wbudowaną konsolę administracyjną umożliwiającą wykonywanie zadań administracyjnych przez przeglądarkę internetową.
26. Narzędzie musi zawierać wbudowane narzędzia umożliwiające administrację i konfigurację platformy z poziomu linii poleceń działające na Microsoft Windows oraz w systemach z rodziny Linux.
27. Narzędzie musi zawierać wbudowany interfejs programistyczny API dostępny przez protokół REST umożliwiający administrację platformą przy użyciu narzędzi zewnętrznych.
28. Narzędzie musi umożliwiać integrację z potokami CI/CD platformy DevOps GitLab zamawiającego w zakresie budowania oraz wdrażania skonteneryzowanych aplikacji.
29. Narzędzie musi umożliwiać instalację sterowników sieciowych Kubernetes CNI (Container Network Interface) pochodzących od różnych dostawców.
30. Narzędzie musi zawierać wbudowaną wewnętrzną wirtualną sieć (SDN) umożliwiającą komunikację pomiędzy aplikacjami i usługami uruchomionymi na platformie oraz dwukierunkową komunikację na zewnątrz.
31. 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.
32. 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.
33. 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.
34. Narzędzie (Ingress router) musi pozwalać na terminację SSL, reekrypcję SSL oraz przekazanie połączenia SSL bezpośrednio do kontenera.
35. Narzędzie musi umożliwiać komunikację SSL w sieci wewnętrznej pomiędzy wybranymi usługami bez konieczności implementacji logiki komunikacji SSL w poszczególnych usługach.

36. Narzędzie musi umożliwiać szyfrowanie komunikacji w sieci wewnętrznej pomiędzy węzłami klastra przy użyciu IPSec lub TLS.
37. 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.
38. Narzędzie musi posiadać moduł umożliwiający balansowanie obciążenia poszczególnych węzłów klastra w celu optymalizacji konsumpcji zasobów.
39. Narzędzie musi pozwalać na udostępnienie komponentów aplikacji z wykorzystaniem usługi loadbalancera L4 stosu TCP/IP Kubernetes typu NodePort.
40. Narzędzie musi umożliwiać instalację sterowników Kubernetes CSI (Container Storage Interface).
41. Narzędzie musi pozwalać na uruchamianie aplikacji stanowych, które zapisują i odczytują dane z trwałego nośnika poprzez następujące interfejsy: NFS, Ceph RDB, CephFS, iSCSi, Fibre Channel, VMWare VMDK.
42. Narzędzie musi pozwalać na wykorzystywanie przez aplikacje stanowe lokalnych zasobów dyskowych znajdujących się na węzłach klastra takich jak lokalne dyski, partycje i urządzenia blokowe.
43. Narzędzie musi mieć możliwość tworzenia wolumenów Kubernetes (Persistent Volume) z trybem dostępu RWO i RWX.
44. Narzędzie musi umożliwiać tworzenie obiektów Storage Class pozwalających na dynamiczne tworzenie i alokacje wolumenów Kubernetes.
45. Narzędzie musi umożliwiać tworzenie wolumenów Kubernetes typu blokowego i plikowego.
46. W przypadku wykorzystania rozwiązań SDS (Software-defined storage), narzędzie musi zapewniać niezawodność przechowywanych danych poprzez posiadanie dwóch lub więcej replik danych oraz erasurecoding.
47. Narzędzie musi umożliwiać uruchomienie mechanizmu śledzenia komunikacji pomiędzy usługami uruchomionymi na platformie zgodnej z OpenTracing API.
48. Narzędzie musi posiadać mechanizmy agregacji logów zarówno w zakresie komponentów płaszczyzny sterowania, węzłów obciążeniowych klastrów, jak i uruchamianych aplikacji.
49. Narzędzie musi posiadać wbudowany mechanizm umożliwiający przesyłanie logów do zewnętrznych systemów agregacji i analizy logów takich jak Splunk, ElasticSearch, Fluentd, Syslog, Loki.

50. 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.
51. Narzędzie musi zawierać mechanizm alertowania i wysyłania powiadomień w przypadku wygenerowania alertów.
52. Narzędzie musi posiadać możliwość tworzenia kopii zapasowych komponentów płaszczyzny sterowania klastrów tj. bazy etcd oraz uruchamianych w ich obrębie aplikacji.
53. Narzędzie musi umożliwiać tworzenie kopii zapasowych woluminów danych aplikacji stanowych niezależnie od wykorzystywanego sterownika CSI.
54. Narzędzie musi posiadać mechanizmy umożliwiające migrację aplikacji (konfiguracji i danych) pomiędzy różnymi klastrami.
55. Nadrzędnie musi zapewniać mechanizmy automatyzujące tworzenie oraz retencje kopii zapasowych.
56. Nadrzędnie nie może wykluczać użycia zewnętrznych systemów tworzenia kopii zapasowych tj. Commvault, Veeam.
57. Narzędzie musi zawierać wbudowane mechanizmy uwierzytelniania i autoryzacji użytkowników oparte na OAuth 2.0, oraz umożliwiać konfigurację dostępu opartego na rolach dla różnych grup użytkowników w tym administratorów i programistów.
58. Narzędzie musi pozwalać na integrację z zewnętrznymi bazami użytkowników w tym Microsoft Active Directory oraz serwerami autoryzacji zgodnymi z OAuth 2.0.
59. Narzędzie musi pozwalać na definiowanie uprawnień do zarządzania oddzielnie różnymi klastrami i grupami klastrów.
60. Narzędzie musi umożliwiać definiowanie różnych projektów dla poszczególnych aplikacji i przypisywania uprawnień do nich dla określonych grup użytkowników.
61. Narzędzie musi zawierać wbudowany mechanizm umożliwiający administratorom określenie uprawnień dla uruchamianych na platformie kontenerów tj. uprawnienia użytkownika, dostępu do zasobów systemowych oraz profile seccomp.
62. Narzędzie musi umożliwiać przechowywanie konfiguracji klastra i aplikacji Kubernetes w formie zaszyfrowanej.
63. Narzędzie musi zawierać elastyczny silnik polityk, który umożliwia definiowanie i egzekwowanie polityk konfiguracji platformy i aplikacji wdrożonych na platformie.

64. 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.
65. Narzędzie musi posiadać mechanizmy do zablokowania uruchomienia kontenera z obrazu, pochodzącego z nieznanych źródeł, niespełniającego przyjętych polityk lub zawierającego lukę bezpieczeństwa.
66. Narzędzie musi umożliwiać śledzenie procesów uruchomionych w kontenerach i wykrywanie aktywności niezgodnych ze zdefiniowanymi politykami bezpieczeństwa.
67. Narzędzie musi umożliwiać ciągłe skanowanie obrazów w celu wykrycia znanych podatności w bibliotekach systemowych oraz aplikacyjnych uruchamiane automatycznie w określonych interwałach czasowych.
68. 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.
69. Narzędzie musi umożliwiać analizę uprawnień Kubernetes opartych na rolach (RBAC) przypisanych do użytkowników i kont serwisowych (service accounts).
70. Narzędzie musi umożliwiać śledzenie zdarzeń w klastrze zapisywanych w Kubernetes audit log w celu wykrywania niezgodności z politykami bezpieczeństwa.
71. Narzędzie musi zawierać wbudowany dedykowany moduł do zarządzania i monitorowania komunikacji sieciowej dla aplikacji zbudowanych w architekturze mikro usług.
72. Narzędzie musi umożliwiać automatyczne generowanie polityk sieciowych na podstawie śledzenia ruchu i polityk bezpieczeństwa.
73. 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.
74. Narzędzie do zarządzania komunikacją sieciową musi posiadać wbudowaną konsolę webową umożliwiającą konfigurację i wizualizację komunikacji wewnątrz Service Mesh.
75. Narzędzie nie może wykluczać zastosowania ochrony uruchamianych aplikacji w postaci zewnętrznych rozwiązań WAF (Web Application Firewall).
76. Narzędzie musi udostępniać warstwę raportową dot. zarządzania podatnościami, zgodności z przyjętymi standardami i regulacjami oraz umożliwiać eksport tych raportów do oddzielnych plików.

77. Narzędzie musi umożliwiać filtrowanie informacji o zgodności z poszczególnymi standardami i regulacjami na poziomie klastra, węzłów klastra lub projektów (namespaces).
78. W ramach przedmiotu zamówienia Wykonawca zapewni wsparcie techniczne producenta na okres 36 miesięcy, świadczone według poniżej opisanych minimalnych wymagań:
    1) Dostępu do aktualizacji, poprawek oraz narzędzi potrzebnych do utrzymania działania Platformy,
    2) Dostępu do materiałów takich jak: techniczna dokumentacja, internetowa baza wiedzy, forum internetowe dotyczące Platformy
    3) Zgłaszania nieograniczonej liczby błędów w działaniu Platformy o niżej wymienionych priorytetach:
        a) błąd pilny, oznacza że Platforma nie działa i żadne obejście nie jest od razu dostępne,
        b) błąd wysoki, oznacza że główna funkcjonalność Platformy jest poważnie ograniczona, dostępne są tymczasowe obejścia problemu,
        c) błąd średni, oznacza że nastąpiła częściowa, niekrytyczna utrata funkcjonalności Platformy,
        d) błąd niski, oznacza że występują drobne, nie mające większego wpływu na działanie Platformy błędy.
    4) Obsługi nieograniczonej liczby zgłoszeń błędów, wad, awarii oraz konsultacji w zakresie konfiguracji, funkcjonalności, obsługi platformy kontenerowej Kubernetes do zarządzania klastrami świadczone w dni robocze w godz. 9.00 - 17.00 w siedzibie Zamawiającego lub zdalnie poprzez bezpieczne szyfrowane połączenie,
    5) 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 6),
    6) Zgłoszenia przyjmowane będą przez Wykonawcę pod podany w umowie:
        a) nr telefonu,
        b) adresem e-mail,
        c) serwisem WWW, udostępniony przez Wykonawcę internetowy system zgłoszeń.
    7) Wykonawca niezwłocznie po otrzymaniu zgłoszenia przystąpi do jego rozpoznania i podejmie działania zmierzające do rozwiązania, zgodnie z przyjętym priorytetem zgłoszenia:
        a) błąd pilny – nastąpi maksymalnie do 1 godziny roboczej,
        b) błąd wysoki – nastąpi maksymalnie do 4 godzin roboczych,
        c) błąd średni – nastąpi maksymalnie do 8 godzin roboczych,

d) błąd niski - nastąpi maksymalnie do 16 godzin roboczych.

8) Wykonawca zapewni nadzorowanie zgłoszeń eskalowanych do producenta.

79. Wykonawca zapewni dodatkowe wsparcie techniczne na okres 36 miesięcy, świadczone według poniżej opisanych minimalnych wymagań:

1) wsparcie technicznego w zakresie: Analizy, wykonania i wsparcia w konfiguracji elementów platformy kontenerowej Kubernetes do zarządzania klastrami;
2) wsparcie w analizie i konfiguracji elementów platformy odpowiedzialnych za warstwy bezpieczeństwa, sieci i aplikacji;
3) wsparcie w analizie i konfiguracji rozwiązań zintegrowanych systemu odpowiedzialnych za procesy CI/CD, katalog usług, podsystemy metryk, agregacji logów, persystencji danych;
4) wsparcie w analizie i konfiguracji rozwiązań RBAC, separacji przestrzeni nazw;
5) wsparcie w analizie i konfiguracji bezprzerwowych procesów aktualizacji.