



Opis wdrożenia Platformy Technologicznej epodreczniki.pl na zasobach Poznańskiego Centrum Superkomputerowo-Sieciowego

w ramach realizacji umowy “pomostowej”
nr 427/PCSS/2016

Poznań, 21 lutego 2017 r.



Spis treści

Wstęp	3
Opis ogólny konfiguracji Platformy na zasobach PCSS	3
Schemat logiczny	3
Podstawowe informacje	5
Szczegółowe informacje o konfiguracji wybranych komponentów	5
Reverse Proxy, Cache, Load Balancer	5
Portal	6
Usługi backendowe	7

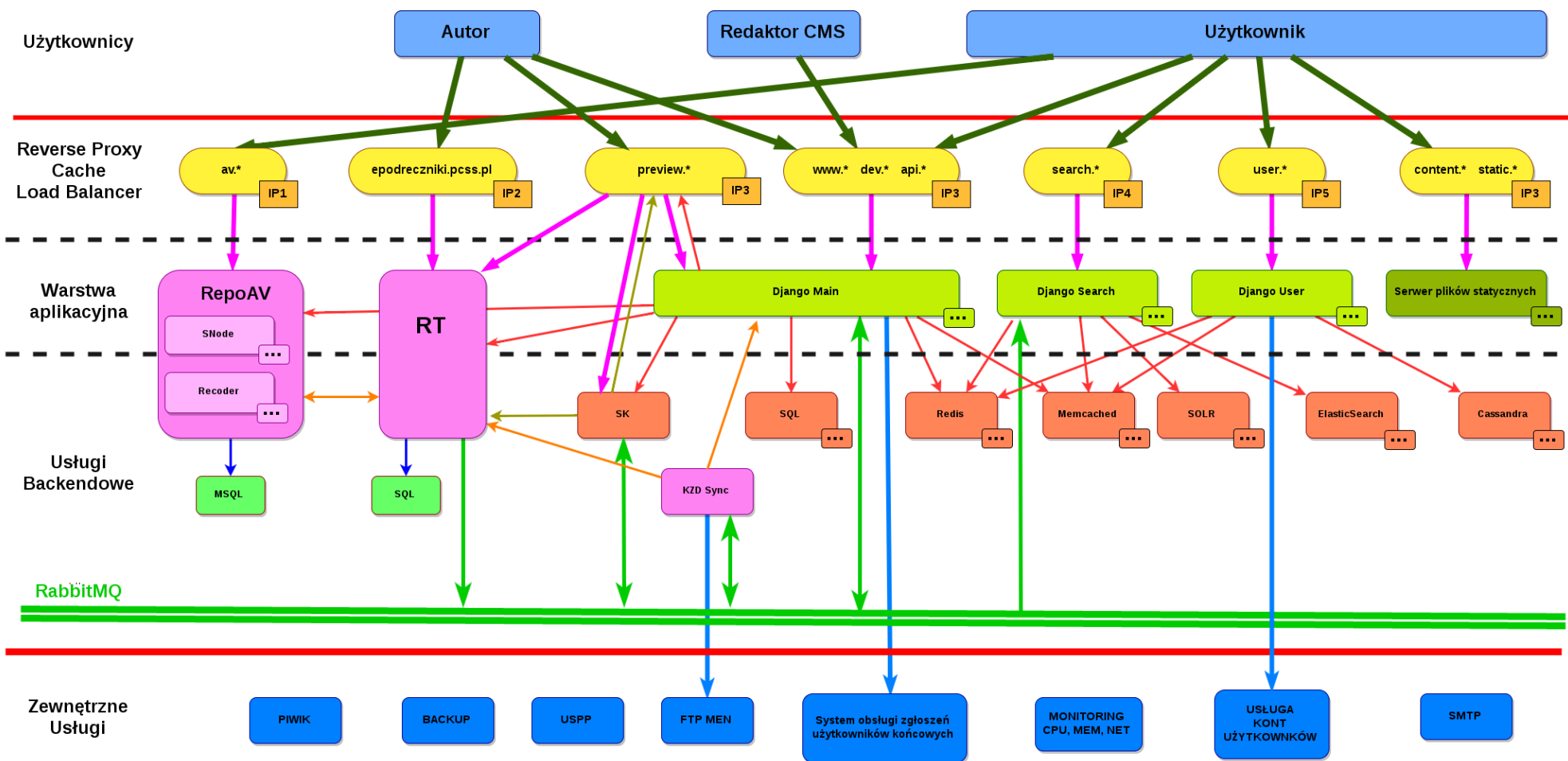
Wstęp

Niniejszy dokument opisuje strukturę logiczną komponentów Platformy Technologicznej epodreczniki.pl wdrożonej na zasobach Poznańskiego Centrum Superkomputerowo-Sieciowego (PCSS) na potrzeby realizacji umowy “pomostowej” z Ośrodkiem Rozwoju Edukacji (ORE) według stanu na dzień 21 lutego 2017 r. Wdrożenie jest oparte o dokumentację techniczną platformy, w szczególności o “Dokumentację architektury”, przekazaną ORE w formie archiwum o sumie kontrolnej 2c67ba72924b876a8737180b7de9aeded51e6604 wraz z aktualizacją o sumie kontrolnej 100f15807ce084fe964c0767bb833303ee0483bd. Celem dokumentu nie jest uzupełnianie dokumentacji technicznej, ale przekazanie informacji o sposobie realizacji umowy pomostowej, w tym utrzymywania systemu informatycznego dużej skali zapewniającego wysoką dostępność krytycznych elementów platformy oraz o wykorzystaniu infrastruktury krytycznej i centrum danych PCSS w kontekście świadczenia usługi utrzymania platformy epodreczniki.pl na rzecz ORE. Dokument opisuje strukturę logiczną platformy wdrożonej na zasobach PCSS w stopniu wystarczającym do zrozumienia działania tej instancji platformy, powiązań i interakcji pomiędzy jej komponentami, a jednocześnie nienaruszającym bezpieczeństwa infrastruktury krytycznej PCSS.

Opis ogólny konfiguracji Platformy na zasobach PCSS

Schemat logiczny

Poniższy diagram (Rysunek 1) przedstawia logiczną strukturę komponentów Platformy wraz ze schematem odwołań pomiędzy nimi oraz z podziałem na wyodrębnione warstwy. Poniższa struktura omówiona została w kolejnych rozdziałach dokumentu.



Rysunek 1. Diagram przedstawiający logiczną strukturę komponentów i powiązań Platformy.

Bloki w warstwach: aplikacyjnej, usług backendowych i usług zewnętrznych, a także szyna "RabbitMQ" odpowiadają logicznym komponentom infrastruktury.

Strzałki oznaczają, że połączone nimi komponenty komunikują się ze sobą, a **kierunek strzałek** odpowiada standardowemu kierunkowi inicjowania danej komunikacji.

Wielokropek przy komponencie oznacza, że dany komponent w celu zachowania wysokiej dostępności utrzymywany jest na **co najmniej dwóch węzłach**.

Podstawowe informacje

Domeny publiczne Platformy są w ramach instancji Platformy w PCSS utrzymywane na 5 logicznych grupach publicznych adresów IP, które są przypisane do domen w sposób wskazany na schemacie (oznaczenie "IP" + numer grupy). Publiczne adresy IP są przyporządkowane do węzłów w warstwie "Reverse Proxy, Cache, Load Balancer". Zapytania HTTP/HTTPS do tych domen są dalej kierowane do odpowiednich węzłów z komponentami platformy, którymi – w zależności od domeny – są: RepoAV, RT, SK, aplikacje Django Portalu oraz serwery plików statycznych Portalu. Pozostałe komponenty stanowią odseparowany zestaw usług i nie obsługują bezpośrednio zapytań HTTP/HTTPS od użytkowników. Usługami typu Backend dla komponentów Platformy są w instancji PCSS również usługi niewchodzące w skład oprogramowania Platformy, które wyróżniono na schemacie. Usługi te utrzymywane są zarówno w PCSS, jak i poza infrastrukturą oraz kontrolą PCSS (serwer FTP działający na zasobach MEN). Dla zachowania odpowiedniej czytelności schematu nie dodano strzałek do usług zewnętrznych, które komunikują się z wieloma komponentami Platformy, jak np. backup czy monitoring.

Szczegółowe informacje o konfiguracji wybranych komponentów

Reverse Proxy, Cache, Load Balancer

Warstwa ta jest bezpośrednio związana z infrastrukturą krytyczną PCSS i pośredniczy w przekazywaniu ruchu sieciowego między siecią szkieletową a komponentami realizującymi funkcjonalność Platformy. Jest ona złożona ze standardowych rozwiązań pozwalających na optymalizację aplikacji internetowych (m.in. reverse proxy, cache, load balancer) i jest włączona bezpośrednio w routery brzegowe sieci szkieletowej. Warstwa ta separuje usługi warstwy aplikacyjnej i backendowej od sieci Internet i użytkowników końcowych. W warstwie tej realizowane jest również szyfrowanie SSL/TLS na potrzeby komunikacji HTTPS.

Ponadto, dla publicznej subdomeny *preview* oraz dla wewnętrznej subdomeny *repo* warstwa ta realizuje mechanizm optymalizacji wydajności powszechnie znany jako X-Sendfile w odniesieniu do plików pobieranych z komponentów RT i SK. Dodatkowo, ze względu na

konieczność czyszczenia pamięci cache utrzymywanej w tej warstwie, ma miejsce komunikacja z warstwy niższej (portali Django) do tej warstwy. W celu zrealizowania tych dwóch funkcjonalności (X-Sendfile i unieważnianie zasobów pamięci cache), w ramach wdrożenia Platformy na zasobach PCSS, dokonano specyficznego dla infrastruktury PCSS dostosowania kodu aplikacyjnego Django (w miejscu wskazanym przez ustawienie: `SURROUND_HTTPCACHE_IMPLEMENTATION_MODULE` w konfiguracji Django).

Portal

W instancji Platformy wdrożonej na zasobach PCSS węzły z aplikacją Django Portalu podzielono na 3 grupy logiczne, nazwane *main*, *search* i *user*, które zostały przeznaczone do obsługi różnych subdomen, jak przedstawiono na schemacie. Wszystkie węzły z każdej z tych grup zawierają ten sam kod źródłowy i mają tę samą konfigurację. Zapytania HTTP są kierowane do odpowiednich grup na podstawie subdomen, które są obsługiwane przez różne publiczne adresy IP.

Katalog `epodreczniki-portal/local/` jest współdzielony pomiędzy wszystkimi węzłami z aplikacją Django Portalu. Katalog `epodreczniki-portal/portal/static/repository/content/` jest współdzielony pomiędzy wszystkimi węzłami Portalu, w tym także serwerami plików statycznych.

Zapytania do subdomen *content* i *static* są obsługiwane przez odrębne węzły, na których uruchomione są serwery HTTP udostępniające pliki statyczne Portalu. Węzły te udostępniają następujące katalogi ze struktury katalogów Portalu:

- katalog `epodreczniki-portal/portal/static/repository/content/` – pliki z tego współdzielonego katalogu są dostępne pod `http://content.<TOP_DOMAIN>/content/`
- katalogi `epodreczniki-portal/portal/collected-static/` i `epodreczniki-portal/portal/static/` – pliki z tych katalogów są dostępne pod `http://static.<TOP_DOMAIN>/portal/`
- katalog `epodreczniki-portal/portal/static/3rdparty/` – pliki z tego katalogu są dostępne pod `http://content.<TOP_DOMAIN>/global/libraries/` oraz pod `http://static.<TOP_DOMAIN>/portal/3rdparty/`
- katalog `epodreczniki-portal/portal/static/repository/resources/` – pliki z tego katalogu są dostępne pod `http://content.<TOP_DOMAIN>/resources/`

Czerwone strzałki na schemacie pomiędzy daną grupą logiczną węzłów Django a usługami backendowymi oznaczają, że dana grupa logiczna węzłów Django wykorzystuje połączenie z daną usługą backendową do bieżącej obsługi zapytań HTTP. Jednakże, ze względu na osadzenie tego samego kodu źródłowego i tej samej konfiguracji Portalu na wszystkich



węzłach wszystkich grup logicznych, faktycznie istnieją połączenia pomiędzy każdym węzłem Django a wszystkimi usługami backendowymi, lecz nie są one używane do bieżącej obsługi zapytań tam, gdzie nie zostały naniesione na schemat. Zaletą takiego rozwiązania jest możliwość szybkiego przejęcia funkcji realizowanych w danej grupie logicznej przez węzeł Django z innej grupy logicznej, np. w razie awarii.

Usługi backendowe

W swojej instalacji PCSS zastosował produkcyjne rozwiązania informatyczne dla zachowania wysokiej dostępności i wydajności krytycznych komponentów. Komponenty: Cassandra, Redis, Memcached, Elasticsearch są wdrożone w PCSS jako wielowęzłowe klastry. Komponenty: SQL Portalu, RabbitMQ i Solr są wdrożone w trybie master-slave.