



# SISTEME TOLERANTE LA DEFECTE

## Tema #2 Site web cu chat peste Kubernetes

Termen de predare: Ultimul laborator

### Obiective

Scopul acestei teme este de a implementa, folosindu-vă de Kubernetes si de diferite tehnologii pentru backend, baza de date și frontend, un site web ce conține și un chat.

### Enunț

Această temă este personalizată pentru **Buzatu C. Constantin-Mihai**.

Obiectivul temei este de a crea un website ce conține o aplicație de chat. Tema va fi implementată folosind mai multe deployment-uri peste Kubernetes. Arhitectura acestei aplicații va cuprinde mai multe elemente:

1. Site-ul web efectiv va fi ținut pe un content management system (Wordpress; Joomla; Drupal). Acestei teme îi este asignată **Wordpress** cu **3** replici. Site-ul va fi expus pe portul default (80). Puteți folosi webgui-ul pentru a construi un mic site pentru un magazin, o pizzerie, alegerea e a voastra (nu pierdeți mult timp dar nici nu lăsați gol).

2. Sistemul de CMS va folosi o bază de date proprie, pe aceasta o puteți identifica în documentația de pe Docker Hub.

3. Sistemul de chat va fi introdus în pagina web folosind un iframe html. Codul pentru acesta se va afla pe un server web (Php + Apache; Python + Django sau Flask; Node.js). Acestei teme îi este asignată folosirea **Php+Apache** cu **4** replici. Chat-ul va fi expus pe portul 88.

4. Mesajele din chat vor fi stocate într-o bază de date (MongoDB; MySQL; MariaDB; Postgres; CouchDB; Redis). Acestei teme îi este asignată folosirea **CouchDB**.

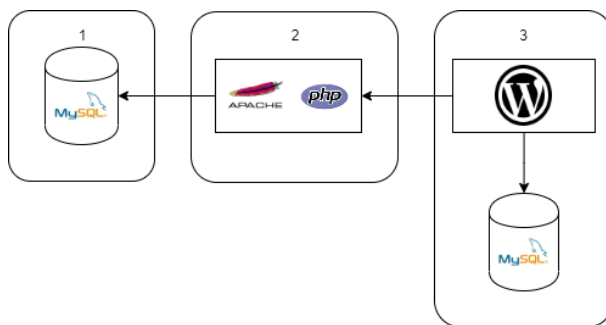


Figure 1: Exemplu de arhitectura pentru aplicatia in discutie

Figura ?? prezintă o arhitectura în care baza de date aleasă este **MySQL**, backend-ul este alcătuit din **Php + Apache**, iar layer-ul de prezentare este bazat pe **Wordpress**, conectat la o baza de date **MySQL**.



## Detalii implementare

Pentru majoritatea componentelor este recomandat să folosiți containere de pe Docker hub. În general le puteți folosi cu doar mici modificări. Pentru containerul care ține serverul web ce oferă pagina de chat va trebui să implementați voi codul și să îl adăugați la containerul aferent. Codul va fi implementat în limbajul ales de voi care apare mai sus în acest document.

Tema va consta dintr-un fișier de tip `.yaml` din care va putea fi pornită prin comanda `apply` întreaga arhitectură. O dată dat `apply` totul va trebui să funcționeze. **Când prezentați nu va fi permis să faceți nici o modificare după ce dați `apply`, nici măcar la CMS.**

Pentru fiecare componentă se va crea un folder în care se va afla fișierul `Dockerfile` alături de oricare alte fișiere (cod, export bază de date, șamd).

Recomandare implementare chat:

1. Pentru stocarea mesajelor în baza de date, se vor salva următoarele: Mesajul în format text, ASCII; Timestamp-ul trimiterii acestui mesaj.
2. Pagina de chat va conține un formular în care se introduce un mesaj text, și va avea un buton de send. Mesajele din baza de date vor fi afișate deasupra acestui formular.
3. Chatul va fi introdus în pagina principală a site-ului printr-un `iframe`. Pentru a conecta backend-ul cu layer-ul de prezentare, indiferent de tehnologia folosită pentru aceasta ultimă componentă, veți folosi un element de tip **`iframe`** pentru a afișa pagina primită de la backend.

## Prezentare și punctare

În ziua prezentării, mașinile Kubernetes (două) trebuie să fie pregătite și pornite. La fel pentru toate add-on-urile (de ex registry). Containerele ce necesită acest lucru trebuie să fie deja build-uite și puse în registry.

În momentul prezentării va trebui să:

- arătați că nu e nici un obiect creat pe cluster;
- aplicați `.yaml`-ul pe cluster;
- intrați pe site și arătați că acesta funcționează, că chat-ul merge;
- **NU** aveți voie să faceți nici o configurație la nici una din componente (nici cms).

Distribuția punctajului este următoarea:

- 40 puncte - Chat-ul funcționează conform specificațiilor;
- 20 puncte - Site-ul integrează tot și funcționează conform specificațiilor;
- 30 puncte - Veți primi două-trei instrucțiuni cu operații peste cluster din linia de comandă; Punctajul se primește doar dacă este evident că știți ce faceți și dați comenziile rapid. Va exista un timer. Acest punctaj se dă doar dacă tema funcționează.

O arhivă `.zip` cu toate fișierele va fi pusă la dispoziție pentru verificări plagiat.