

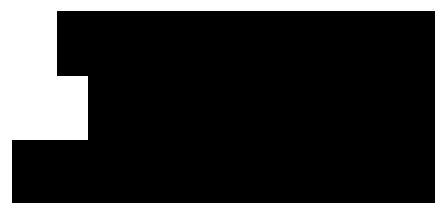
UNIVERSITATEA "SPIRU HARET" – BUCURESTI  
FACULTATEA DE INGINERIE SI INFORMATICA

INVATAMANT CU FRECVENTA

**PIE – PROIECT INFORMATIC ECHIPA**

**Kubernetes in Cloud**

ROSCULETE DRAGOS-CRISTIAN



ANUL III, GRUPA 304

Conf. Univ. Dr. Ing. [REDACTED]

# Kubernetes

## 1. Kubernetes Overview

Kubernetes, also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications.

<https://kubernetes.io/>

**Container deployment era:** Containers are similar to VMs, but they have relaxed isolation properties to share the Operating System (OS) among the applications. Therefore, containers are considered lightweight. Similar to a VM, a container has its own filesystem, share of CPU, memory, process space, and more. As they are decoupled from the underlying infrastructure, they are portable across clouds and OS distributions.

Containers have become popular because they provide extra benefits, such as:

- Agile application creation and deployment: increased ease and efficiency of container image creation compared to VM image use.
- Continuous development, integration, and deployment: provides for reliable and frequent container image build and deployment with quick and efficient rollbacks (due to image immutability).
- Dev and Ops separation of concerns: create application container images at build/release time rather than deployment time, thereby decoupling applications from infrastructure.
- Observability not only surfaces OS-level information and metrics, but also application health and other signals.
- Environmental consistency across development, testing, and production: Runs the same on a laptop as it does in the cloud.
- Cloud and OS distribution portability: Runs on Ubuntu, RHEL, CoreOS, on-premises, on major public clouds, and anywhere else.
- Application-centric management: Raises the level of abstraction from running an OS on virtual hardware to running an application on an OS using logical resources.
- Loosely coupled, distributed, elastic, liberated micro-services: applications are broken into smaller, independent pieces and can be deployed and managed dynamically – not a monolithic stack running on one big single-purpose machine.
- Resource isolation: predictable application performance.
- Resource utilization: high efficiency and density.

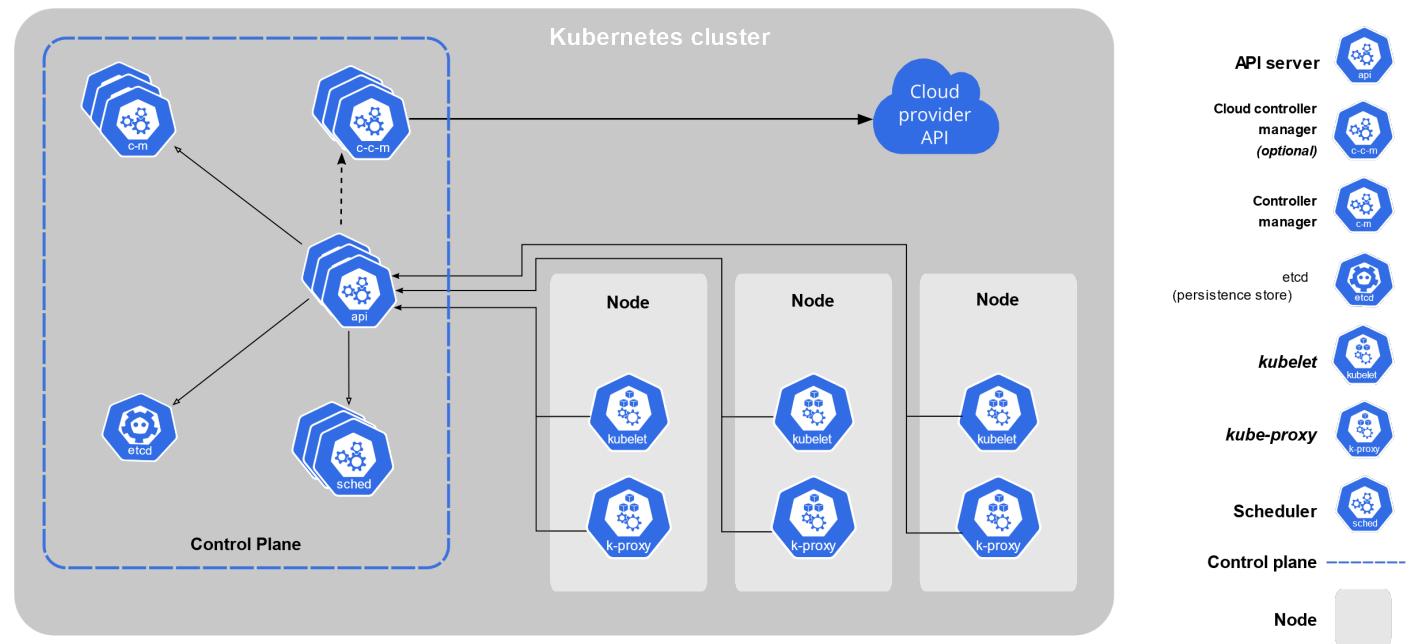
## 2. Kubernetes Components

When you deploy Kubernetes, you get a cluster.

A Kubernetes cluster consists of a set of worker machines, called nodes that run containerized applications. Every cluster has at least one worker node.

The worker node(s) host the Pods that are the components of the application workload. The control plane manages the worker nodes and the Pods in the cluster. In production environments, the control plane usually runs across multiple computers and a cluster usually runs multiple nodes, providing fault-tolerance and high availability.

Here's the diagram of a Kubernetes cluster with all the components tied together.



### Control Plane Components

The control plane's components make global decisions about the cluster (for example, scheduling), as well as detecting and responding to cluster events (for example, starting up a new [pod](#) when a deployment's replicas field is unsatisfied).

Control plane components can be run on any machine in the cluster. However, for simplicity, set up scripts typically start all control plane components on the same machine, and do not run user containers on this machine.

#### **kube-apiserver**

The API server is a component of the Kubernetes [control plane](#) that exposes the Kubernetes API. The API server is the front end for the Kubernetes control plane.

The main implementation of a Kubernetes API server is [kube-apiserver](#). kube-apiserver is designed to scale horizontally—that is, it scales by deploying more instances. You can run several instances of kube-apiserver and balance traffic between those instances.

## **etcd**

Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data. If your Kubernetes cluster uses etcd as its backing store, make sure you have a [back up](#) plan for those data. You can find in-depth information about etcd in the official [documentation](#).

## **kube-scheduler**

Control plane component that watches for newly created [Pods](#) with no assigned [node](#), and selects a node for them to run on.

Factors taken into account for scheduling decisions include: individual and collective resource requirements, hardware/software/policy constraints, affinity and anti-affinity specifications, data locality, inter-workload interference, and deadlines.

## **kube-controller-manager**

Control Plane component that runs [controller](#) processes.

Logically, each [controller](#) is a separate process, but to reduce complexity, they are all compiled into a single binary and run in a single process.

Some types of these controllers are:

- Node controller: Responsible for noticing and responding when nodes go down.
- Job controller: Watches for Job objects that represent one-off tasks, then creates Pods to run those tasks to completion.
- Endpoints controller: Populates the Endpoints object (that is, joins Services & Pods).
- Service Account & Token controllers: Create default accounts and API access tokens for new namespaces.

## **cloud-controller-manager**

A Kubernetes [control plane](#) component that embeds cloud-specific control logic. The cloud controller manager lets you link your cluster into your cloud provider's API, and separates out the components that interact with that cloud platform from components that only interact with your cluster.

The cloud-controller-manager only runs controllers that are specific to your cloud provider. If you are running Kubernetes on your own premises, or in a learning environment inside your own PC, the cluster does not have a cloud controller manager.

As with the kube-controller-manager, the cloud-controller-manager combines several logically independent control loops into a single binary that you run as a single process. You can scale horizontally (run more than one copy) to improve performance or to help tolerate failures.

The following controllers can have cloud provider dependencies:

- Node controller: For checking the cloud provider to determine if a node has been deleted in the cloud after it stops responding
- Route controller: For setting up routes in the underlying cloud infrastructure
- Service controller: For creating, updating and deleting cloud provider load balancers

## Node Components

Node components run on every node, maintaining running pods and providing the Kubernetes runtime environment.

### kubelet

An agent that runs on each [node](#) in the cluster. It makes sure that [containers](#) are running in a [Pod](#).

The kubelet takes a set of PodSpecs that are provided through various mechanisms and ensures that the containers described in those PodSpecs are running and healthy. The kubelet doesn't manage containers which were not created by Kubernetes.

### kube-proxy

kube-proxy is a network proxy that runs on each [node](#) in your cluster, implementing part of the Kubernetes [Service](#) concept.

[kube-proxy](#) maintains network rules on nodes. These network rules allow network communication to your Pods from network sessions inside or outside of your cluster.

kube-proxy uses the operating system packet filtering layer if there is one and it's available. Otherwise, kube-proxy forwards the traffic itself.

## Container runtime

The container runtime is the software that is responsible for running containers.

Kubernetes supports several container runtimes: [Docker](#), [containerd](#), [CRI-O](#), and any implementation of the [Kubernetes CRI \(Container Runtime Interface\)](#).

## Addons

Addons use Kubernetes resources ([DaemonSet](#), [Deployment](#), etc) to implement cluster features. Because these are providing cluster-level features, namespaced resources for addons belong within the kube-system namespace.

## DNS

While the other addons are not strictly required, all Kubernetes clusters should have [cluster DNS](#), as many examples rely on it.

Cluster DNS is a DNS server, in addition to the other DNS server(s) in your environment, which serves DNS records for Kubernetes services.

Containers started by Kubernetes automatically include this DNS server in their DNS searches.

## Web UI (Dashboard)

[Dashboard](#) is a general purpose, web-based UI for Kubernetes clusters. It allows users to manage and troubleshoot applications running in the cluster, as well as the cluster itself.

## Container Resource Monitoring

[Container Resource Monitoring](#) records generic time-series metrics about containers in a central database, and provides a UI for browsing that data.

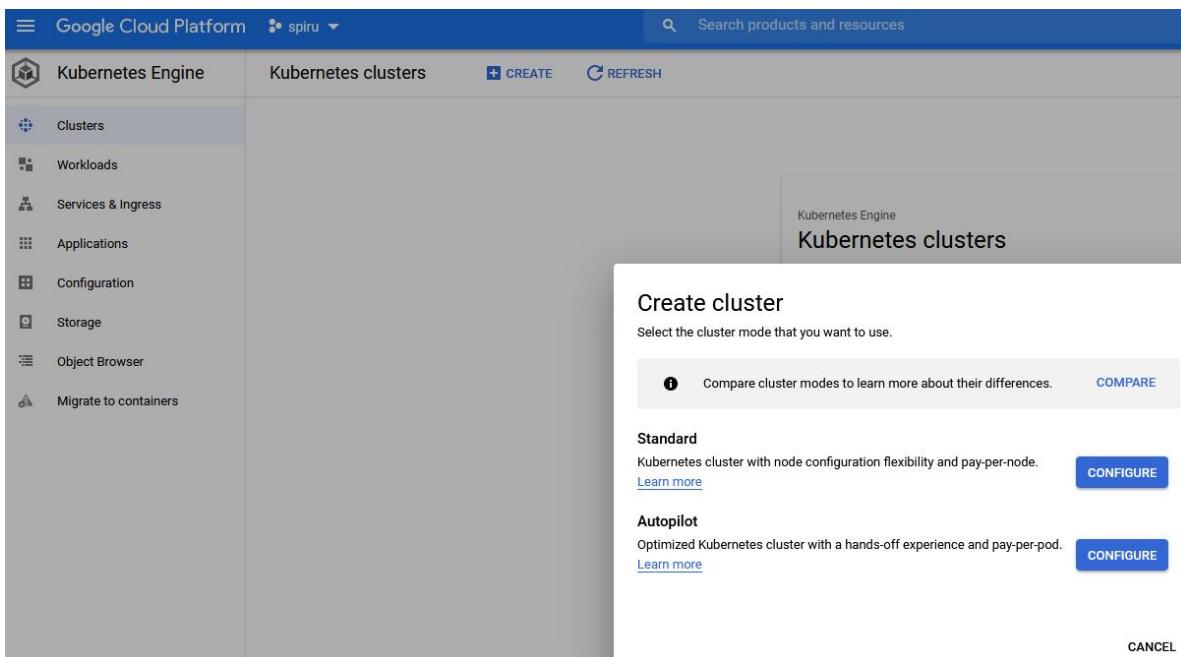
## Cluster-level Logging

A [cluster-level logging](#) mechanism is responsible for saving container logs to a central log store with search/browsing interface.

### 3. Kubernetes Deployment on GCP – Kubernetes ENGINE

#### Prerequisites:

- Have an GCP account: <https://cloud.google.com/apigee/docs/hybrid/v1.1/precog-gcpaccount>
- Create A project and attach billing to it
- Create a GKE <https://cloud.google.com/kubernetes-engine/> autopilot type, use default settings\



- Deploy a sample workload . Click Continue, select cluster and click deploy.

Google Cloud Platform spiru

Kubernetes Engine

Clusters

Workloads

Services & Ingress

Applications

Configuration

Storage

Object Browser

Migrate to containers

Create a deployment

1 Container

Edit container

Existing container image

New container image

Image path \* nginx:latest SELECT

Enter your image path, or choose from Google Container Registry. You can also try to deploy with official nginx image nginx:latest.

Environment variables

+ ADD ENVIRONMENT VARIABLE

Initial command

Overrides the default entrypoint of the container image.

CANCEL    DONE

ADD CONTAINER

CONTINUE

2 Configuration

The screenshot shows the 'Create a deployment' wizard in the Google Cloud Platform Kubernetes Engine. The 'Container' step is currently selected. In the 'Edit container' section, the 'Existing container image' radio button is selected, and the image path 'nginx:latest' is entered in the input field. Below this, a note says 'Enter your image path, or choose from Google Container Registry. You can also try to deploy with official nginx image nginx:latest.' The 'Environment variables' section has a '+ ADD ENVIRONMENT VARIABLE' button. An 'Initial command' input field contains the placeholder text 'Overrides the default entrypoint of the container image.' At the bottom of the container step, there are 'CANCEL' and 'DONE' buttons. Below the container step, the 'Configuration' step is partially visible. The left sidebar lists various Kubernetes resources: Clusters, Workloads (which is selected), Services & Ingress, Applications, Configuration, Storage, Object Browser, and Migrate to containers.

- Expose application. Once the application is deployed we need to expose it so we can access it from a service. We are going to expose it on NodePort

Click on Actions -> Expose and complete the popup

Google Cloud Platform - spiru - Search products and resources

Kubernetes Engine - Deployment details - nginx-1 - Overview - Refresh - Edit - Delete - Actions - Kubectl

**nginx-1**

OVERVIEW DETAILS REVISION HISTORY EVENTS LOGS NEW YAML

CPU Memory

Expose

Expose a resource's Pods using a Kubernetes Service.

Port mapping

Port \* Target port Protocol

80 TCP

+ ADD PORT MAPPING

Service type Node port

Cluster: spiru Namespace: default Labels: app:nginx-1 Logs: Container logs, Audit logs Replicas: 1 updated, 1 ready, 1 available, 0 unavailable Pod specification: Revision 1, containers: nginx-1

Active revisions

Revision	Name	Status	Summary	Created
1	nginx-1-7744rR886d	OK	nginx-1:nginx:latest	Apr 2

\* indicates required field

CANCEL EXPOSE

- Next step is to allow external users to connect to it. For this we will create an ingress to route the traffic to our service. This will create a LoadBalancer and route the traffic based on path.

Google Cloud Platform - spiru - Search products and resources

Kubernetes Engine - Services & Ingress - Refresh - + CREATE INGRESS - DELETE

Clusters Workloads Services & Ingress Applications Configuration Storage Object Browser Migrate to containers

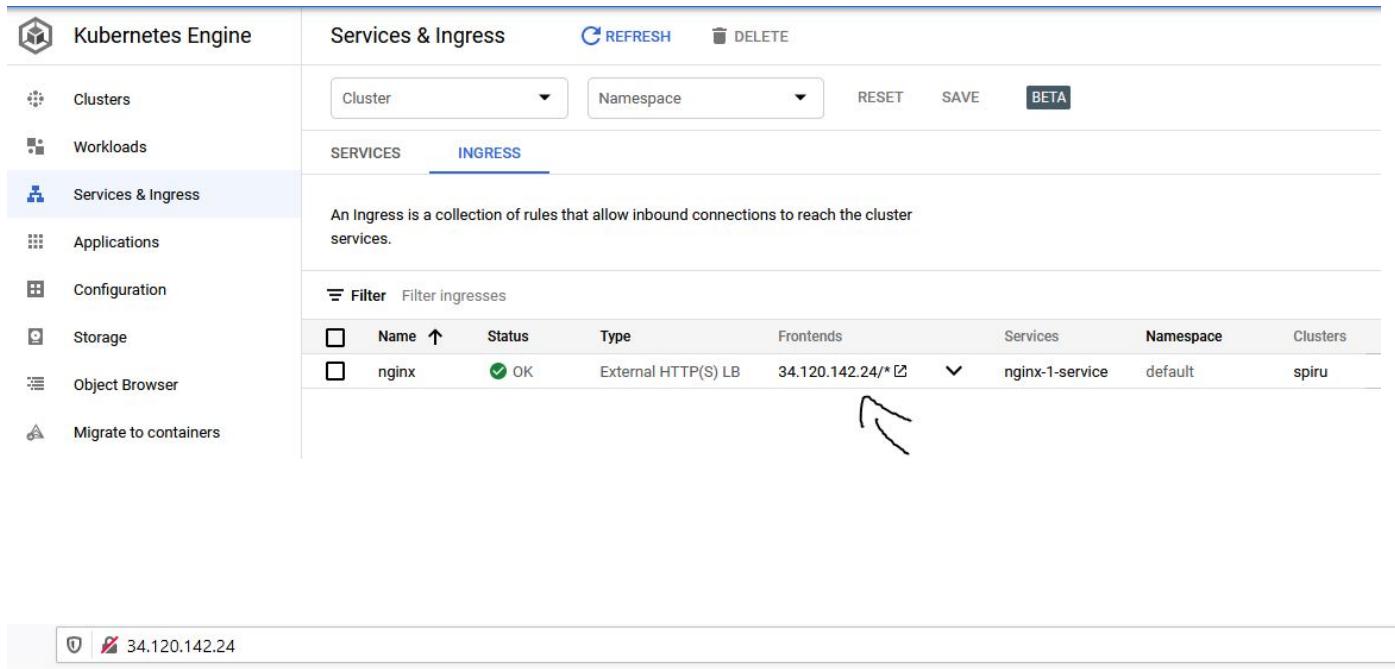
Services INGRESS

Services are sets of Pods with a network endpoint that can be used for discovery and load balancing. Ingresses are collections of rules for routing external HTTP(S) traffic to Services.

Filter Is system object : False Filter services and ingresses

Name	Status	Type	Endpoints	Pods	Namespace	Clusters
nginx-1-service	OK	Node Port	10.80.129.17:80 TCP	1/1	default	spiru

- Once Created we can view the endpoint and access it



The screenshot shows the Kubernetes Engine interface under the Services & Ingress tab. It displays an ingress rule for an Nginx service. The table shows one entry:

<input type="checkbox"/>	Name	Status	Type	Frontends	Services	Namespace	Clusters
<input type="checkbox"/>	nginx	OK	External HTTP(S) LB	34.120.142.24/*	nginx-1-service	default	spira

A hand-drawn arrow points from the bottom right towards the IP address in the browser bar.

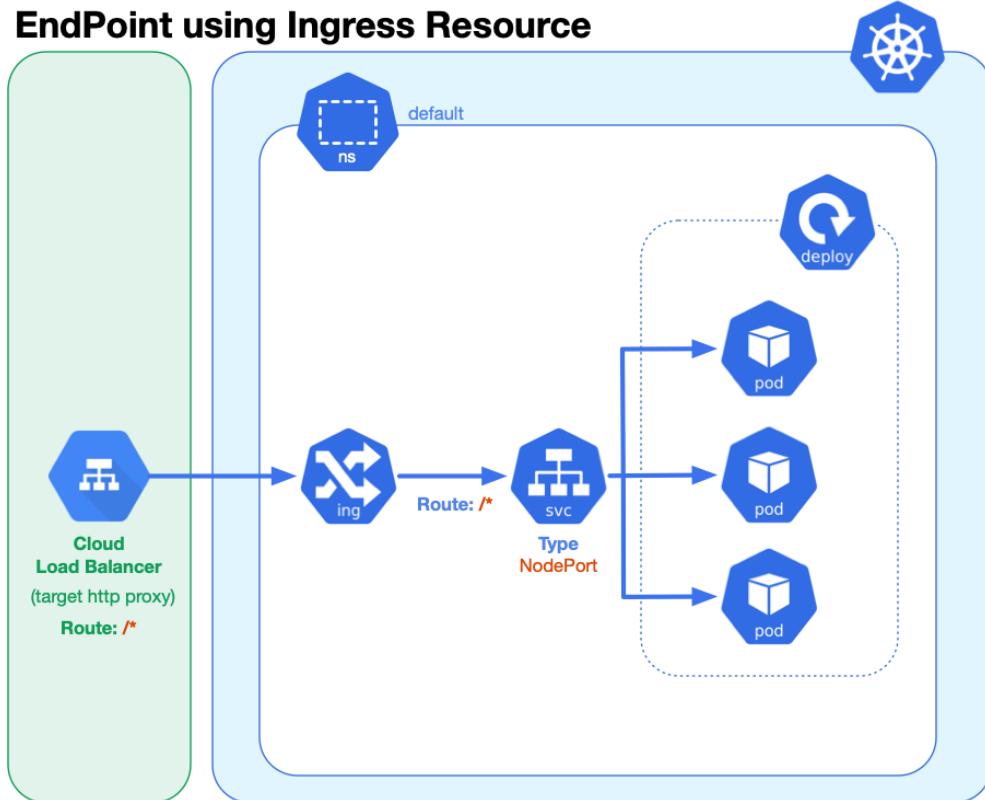
**Welcome to nginx!**

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org). Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

## EndPoint using Ingress Resource



## Deploy image to GCP Docker Registry

Install GCP SDK <https://cloud.google.com/sdk/docs/install>

Once installed auth with your account and also setup docker auth using  
gcloud auth configure-docker

Build and image:

```
build -t eu.gcr.io/spiru-311716/nginx-php .
```

Push and image:

```
docker push eu.gcr.io/spiru-311716/nginx-php
```

# GCP Cloud Source Repository

Source control for your code

Cloud Source Repositories

Create new repository

Repository name\*  
spiru

Project\*  
spiru-311716

OR [Create project](#)

Your repository is billed based on Cloud Source Repositories pricing.

[Cancel](#) [Create](#)

Add code to your repository

Your repository is currently empty. Add some code using a selected method and then refresh your browser. Contents added to this repository can take some time to show up in search results. [Learn more](#).

Select an option to push code to your repository:

Push code from a local Git repository  
 Clone your repository to a local Git repository

Select your preferred authentication method

SSH authentication [Google Cloud SDK](#) Manually generated credentials

1. Install the Google Cloud SDK [.](#)
2. Provide your authentication credentials:  

```
$ gcloud init
```
3. Clone this repository to a local Git repository:  

```
$ gcloud source repos clone spiru --project=spiru-311716
```
- Note: This may display the following message that is safe to ignore:  
"Warning: remote HEAD refers to a nonexistent ref, unable to checkout."
4. Switch to your new local Git repository:  

```
$ cd spiru
```
5. After you've committed code to your local Git repository, push it to this repository:  

```
$ git push -u origin master
```
6. Once you've completed all these steps, refresh your browser.

## Implement CI/CD , Code Build

Build the code on git push, test, create the docker image, push to repo and deploy to Kubernetes Cluster

The CI/CD pipeline is one of the best practices for devops teams to implement, for delivering code changes more frequently and reliably

Continuous integration (CI) and continuous delivery (CD) embody a culture, set of operating principles, and collection of practices that enable application development teams to deliver code changes more frequently and reliably. The implementation is also known as the CI/CD pipeline.

Continuous Deployment takes it a step further and deploys the application automatically at the end of the pipeline process without human intervention .

Components on Google to achieve this:

- Cloud Source Repository

This is where the source code is kept (GIT).

- Code Build

This is an automation tool that based on an event ( like a commit ) can trigger a pipeline . For example build a docker image and

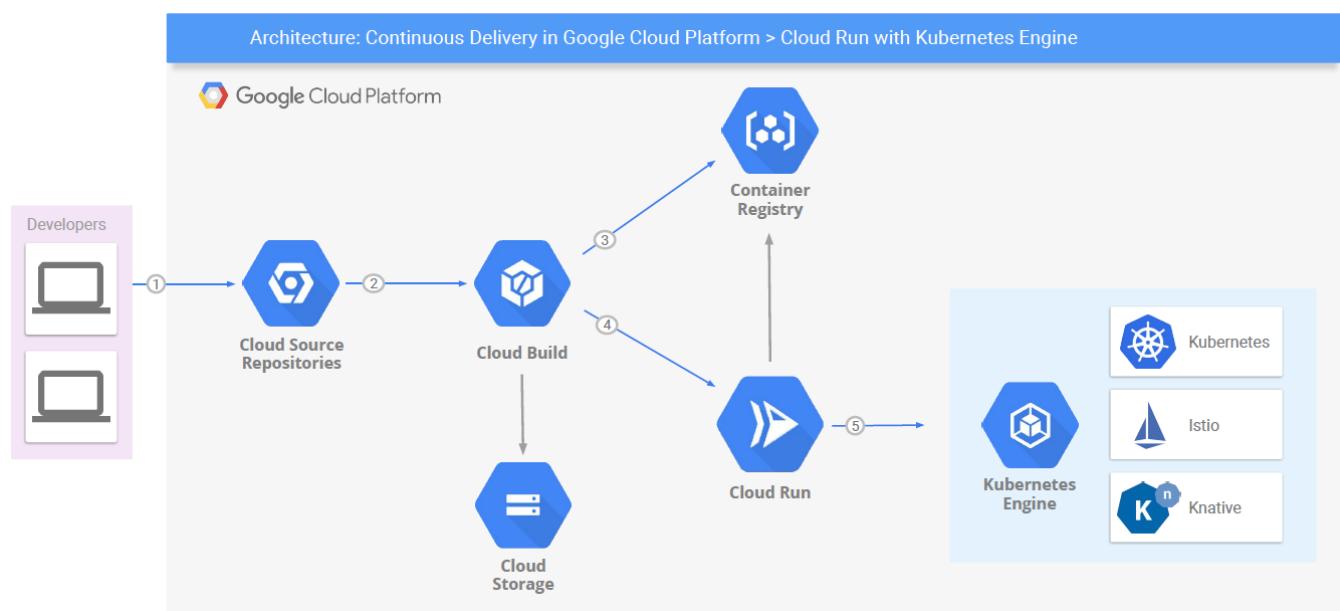
push it to a Docker/Container Registry.

- Container Registry

Here we store all the docker images

- Code Build

We build and deploy the image to Kubernetes



Google Cloud Platform spiru

Cloud Build Triggers Create trigger

Dashboard History Triggers Settings

Name \* nginx  
Trigger name already exists in project

Description nginx

Tags ?

**Event**  
Repository event that invokes trigger  
 Push to a branch  
 Push new tag  
 Pull request (GitHub App only)  
Or in response to  
 Manual invocation  
 Pub/Sub message  
 Webhook event

**Source**  
Repository \* nginx (Cloud Source Repositories)  
Select the repository to watch for events and clone when the trigger is invoked

Branch \* ^master\$  
Use a regular expression to match to a specific branch [Learn more](#)

Invert Regex  
Matches the branch: master

**Configuration**  
Type  
 Cloud Build configuration file (yaml or json)  
 Dockerfile  
Location  
 Repository nginx (Cloud Source Repositories)  
 Inline Write inline YAML

/ Dockerfile directory ?  
The directory will also be used as the Docker build context

Dockerfile name

Release Notes

## Poze project:

Google Cloud Platform - spiru

Kubernetes Engine

Kubernetes clusters

**Clusters**

Name	Location	Mode	Number of nodes	Total vCPUs	Total memory	Notifications	Labels
spiru	europe-central2	Autopilot	-	2.5	4 GB	-	-

**Workloads**

**Services & Ingress**

**Applications**

**Configuration**

**Storage**

**Object Browser**

**Migrate to containers**

Google Cloud Platform - spiru

Kubernetes Engine

Workloads

**Clusters**

**Workloads**

Cluster Namespace RESET SAVE PREVIEW

Workloads are deployable units of computing that can be created and managed in a cluster.

**Filter** Is system object : False Filter workloads

Name	Status	Type	Pods	Namespace	Cluster
catalog	OK	Deployment	1/1	default	spiru
chat	OK	Deployment	1/1	default	spiru
emp-be	OK	Deployment	1/1	default	spiru
emp-fe	OK	Deployment	1/1	default	spiru
nginx-1	OK	Deployment	1/1	default	spiru

**Services & Ingress**

**Applications**

**Configuration**

**Storage**

**Object Browser**

**Migrate to containers**

Google Cloud Platform - spiru

Kubernetes Engine

Services & Ingress

**Clusters**

**Workloads**

**Services & Ingress**

Cluster Namespace RESET SAVE PREVIEW

SERVICES INGRESS

Services are sets of Pods with a network endpoint that can be used for discovery and load balancing. Ingresses are collections of rules for routing external HTTP(S) traffic to Services.

**Filter** Is system object : False Filter services and ingresses

Name	Status	Type	Endpoints	Pods	Namespace	Clusters
catalog-service	OK	Node Port	10.80.131.253:80 TCP	1/1	default	spiru
chat-service	OK	Node Port	10.80.130.141:80 TCP	1/1	default	spiru
emp-be-service	OK	Cluster IP	10.80.129.187	1/1	default	spiru
emp-fe-service	OK	Node Port	10.80.129.136:80 TCP	1/1	default	spiru
nginx-1-service	OK	Node Port	10.80.129.17:80 TCP	1/1	default	spiru

**Applications**

**Configuration**

**Storage**

**Object Browser**

**Migrate to containers**



# Cloud Source Repositories

[My source](#)[All repositories](#)[Project](#)[spiru-311716](#) ▾

## Name

[catalog](#)[chat](#)[emp-backend](#)[emp-frontend](#)[nginx](#)

## Dashboard

[Filter](#) Enter property name or value

### Successful: catalog - automated-deployment-1

Latest Build  
[5/22/21, 2:04 PM](#)Duration  
00:04:06Trigger description  
Build and deploy on push to "master"Source  
[catalog](#)Commit  
[308fb862](#)Build History  
Latest  
 View allAverage Duration [?](#)  
00:03:22Pass - Fail % [?](#)  
100% - 0%

### Successful: chat - automated-deployment-4

Latest Build  
[6/1/21, 10:51 PM](#)Duration  
00:01:04Trigger description  
Build and deploy on push to "master"Source  
[chat](#)Commit  
[5fea819](#)Build History  
Latest  
 View allAverage Duration [?](#)  
00:03:02Pass - Fail % [?](#)  
58% - 42%

### Successful: emp-backend - automated-deployment-3

Latest Build  
[6/2/21, 11:35 PM](#)Duration  
00:01:09Trigger description  
Build and deploy on push to "master"Source  
[emp-backend](#)Commit  
[2ddc2d6](#)Build History  
Latest  
 View allAverage Duration [?](#)  
00:03:45Pass - Fail % [?](#)  
46% - 54%

### Successful: emp-frontend - automated-deployment-2

Latest Build  
[6/3/21, 12:19 AM](#)Duration  
00:04:04Trigger description  
Build and deploy on push to "master"Source  
[emp-frontend](#)Commit  
[d08d785](#)Build History  
Latest  
 View allAverage Duration [?](#)  
00:03:44Pass - Fail % [?](#)  
75% - 25%

### Successful: nginx - nginx

Latest Build  
[5/13/21, 8:29 PM](#)Duration  
00:01:14Trigger description  
nginxSource  
[nginx](#)Commit  
[84799fa](#)Build History  
 View allAverage Duration [?](#)  
00:01:14Pass - Fail % [?](#)  
100% - 0%

# **Chat App**

## **1. Chat App Overview**

Chat App is an web chat application that allows multiple users to connect and create / join chat rooms and talk to each other anonymously. The application does not require user registration or user data, does not store the conversation and the history is lost once the room is closed (all users disconnect from the room). Newly entered users will not see the room chat history (if available). The app can host an unlimited number of rooms and each room can host an unlimited number of users.

The application is built using primarily Node.js and Socket.IO. Socket.IO is a library that enables real-time, bidirectional and event-based communication between the browser and the server.

## **2. Chat App Components**

From the home page the user will be prompted to enter a name and a room name. If the room already exists the user will join that room, if not, the room will be created. There are no unique names for the users, meaning the multiple users can have the exact same name.

Once entered in a room the user is greeted by the host (server) and all other users will be notified by the same host that a new person entered the room.

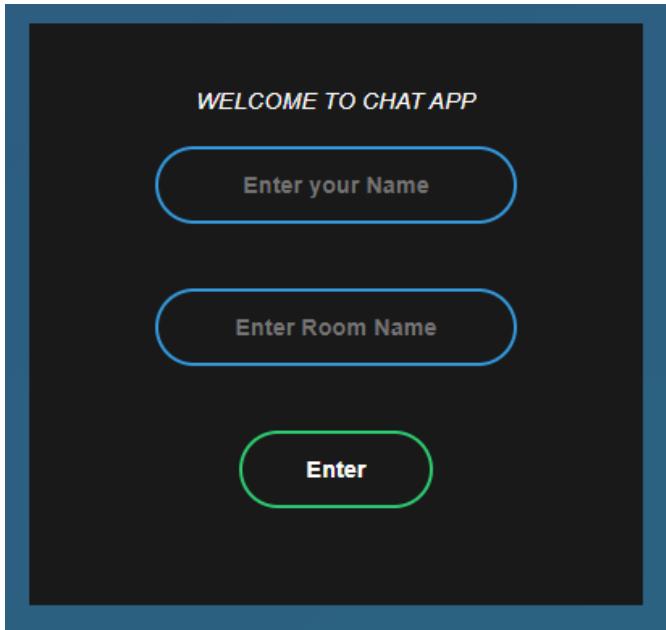
Each message sent by each user will be time stamped and user stamped so that everyone knows who said what and when. Besides this, users can also exchange locations, by clicking the location button if the browser allows it (This feature only works on local and not on prod because it requires https protocol).

The user has two options to quit the chat room:

- a) Close the browser / browser tab
- b) Click on the “Leave room” button from the bottom left side of the screen

In the user clicks on “Leave room” he will be redirect to the home page.

In both instances the host (server) will notify all the other users in the room that the user left. If he was the last person in the room when he left then the room will be also terminated.



Online users

- R. Kavita Ropstar
- Host: 1-17 AM
- Welcome to Sprout!
- George Pavit joined the chat room.

Change Room

Leave Room

Type a message

This screenshot shows the main chat room interface. On the left, there's a sidebar with the title "Online users" and a list containing "R. Kavita Ropstar", "Host: 1-17 AM", "Welcome to Sprout!", and "George Pavit joined the chat room.". Below this are two buttons: "Change Room" and "Leave Room". At the bottom is a text input field with the placeholder "Type a message". The main area is filled with a repeating pattern of small, stylized letters.

Online users

- R. Kavita Ropstar
- Host: 1-17 AM
- Welcome to Sprout!
- George Pavit joined the chat room.
- Host: 1-17 AM
- George Pavit has left Sprout chat room.

Change Room

Leave Room

Type a message

This screenshot shows the main chat room interface. On the left, there's a sidebar with the title "Online users" and a list containing "R. Kavita Ropstar", "Host: 1-17 AM", "Welcome to Sprout!", "George Pavit joined the chat room.", "Host: 1-17 AM", and "George Pavit has left Sprout chat room.". Below this are two buttons: "Change Room" and "Leave Room". At the bottom is a text input field with the placeholder "Type a message". The main area is filled with a repeating pattern of small, stylized letters.

# Product Catalog

Web application, built using the CakePhp 4 framework (php 7.4, MySQL 5.6), bootstrap 4 and jQuery 3

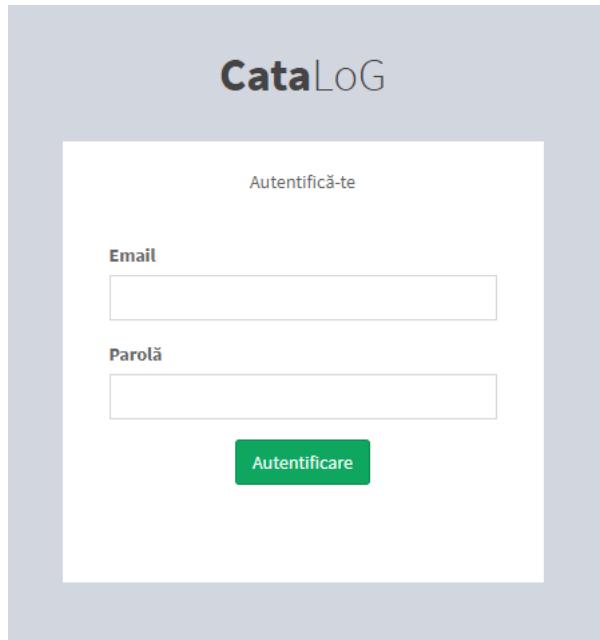
Adding categories and products in a CMS and displaying them in a website.

Creation of a contact form through which users can contact the site administrator

The application is accessible from any device that has a browser installed that supports HTML5 and CSS 3 and has access to the Internet.

- Most of the implementation is done from the terminal, but there are also details that must be solved manually
- CakePhp 4 has its own console, with the help of commands you can automatically generate views, controllers and models, using the relationships between tables in the database
- Creation of the database, having the following tables: users, categories, products, contacts
- Installation of the CakePhp 4 framework through composer
- CMS theme installation – AdminLTE
- Creation of controllers, models and views for CMS
- Creation of controller and front views
- Install image upload plug-in

## Back End



**CataloG**

admin

MAIN NAVIGATION

- Dashboard
- Users
- Products
- Categories
- Contacts

Dashboard

Products Categories Users Contacts

**CataloG**

admin

MAIN NAVIGATION

- Dashboard
- Users**
- Products
- Categories
- Contacts

Users

New

List						Search
<b>Id</b>	<b>Full Name</b>	<b>Email</b>	<b>Is Active</b>	<b>Created</b>	<b>Modified</b>	<b>Actions</b>
9	George	gheorghe.stanescu@gmail.com	1	4/9/21, 11:45 AM	4/9/21, 11:45 AM	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
12	admin	admin@catalo.lo	1	4/14/21, 5:22 PM	4/14/21, 5:22 PM	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

**CataloG**

admin

MAIN NAVIGATION

- Dashboard
- Users
- Products**
- Categories
- Contacts

Products

New

List											Search
<b>Id</b>	<b>Parent</b>	<b>Category</b>	<b>Title</b>	<b>Description</b>	<b>Price</b>	<b>Slug</b>	<b>Featured Image</b>	<b>Is Active</b>	<b>Created</b>	<b>Modified</b>	<b>Actions</b>
1	0	Materiale de construcții	Cărămidă	Lorem ipsum Cărămidă	2022	caramida		✓	4/2/21, 5:40 PM	5/10/21, 1:18 PM	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Gallery</a> <a href="#">Delete</a>
2	1	Piese Auto	test	tessst	994	test		✓	4/8/21, 8:06 AM	4/16/21, 7:14 PM	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Gallery</a> <a href="#">Delete</a>

Categories									<a href="#">New</a>
List <span style="float: right;">Search <input type="text"/> </span>									
ID	Parent	Title	Slug	Featured Image	Description	Is Active	Created	Modified	Actions
1	NA	Materiale de construcții	materiale-de-construcții		Lorem ipsum materiale de construcții	✓	4/2/21, 4:05 PM	4/9/21, 10:11 PM	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
7	Materiale de construcții	tesssst cat	tesssst-cat		tesssst cat Description	✓	4/16/21, 5:25 PM	4/16/21, 5:25 PM	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
2	NA	Piese Auto	piese-auto		Lorem ipsum piese auto	✓	4/2/21, 4:18 PM	4/19/21, 12:57 PM	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
6	Piese Auto	cutie viteze	cutie-viteze		Lorem ipsum cutie viteze	✓	4/11/21, 2:51 PM	4/11/21, 2:51 PM	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
8	Piese Auto	Motoare	motoare		Lorem ipsum motor est	✓	4/19/21, 6:45 PM	4/19/21, 7:04 PM	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
3	NA	test	test		tessst	✓	4/8/21, 8:01 AM	4/8/21, 8:01 AM	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

Contacts									<a href="#">New</a>
List <span style="float: right;">Search <input type="text"/> </span>									
ID	Name	Phone	Message	Created	Actions				
1	test contact	0769171712	test mesaj contact	4/10/21, 9:12 AM	<a href="#">View</a>	<a href="#">Edit</a>	<a href="#">Delete</a>		
2	aaaaaa	01234544888	tesst front	4/10/21, 9:27 AM	<a href="#">View</a>	<a href="#">Edit</a>	<a href="#">Delete</a>		

User Add		<a href="#">Home</a>
Form		
<b>Full Name</b>		
<input type="text"/>		
<b>Email</b>		
<input type="text"/>		
<b>Password</b>		
<input type="password"/>		
<input checked="" type="checkbox"/> Is Active		
<input type="button" value="Submit"/>		

**CataLoG**

MAIN NAVIGATION

- Dashboard
- Users
- Products
- Categories
- Contacts

Product Add

Form

Title	Slug
<input type="text"/>	<input type="text"/>
Parent	Price
<input type="text"/>	<input type="text"/>
Category	Featured Image
<input type="text"/> Materiale de construcții	<input type="button" value="Choose File"/> No file chosen
Keywords	<input type="checkbox"/> Is Active
<b>Description</b>	
<input type="button" value="Normal text"/> <input type="button" value="Bold"/> <input type="button" value="Italic"/> <input type="button" value="Underline"/> <input type="button" value="H1"/> <input type="button" value="H2"/> <input type="button" value="H3"/> <input type="button" value="H4"/> <input type="button" value="H5"/> <input type="button" value="H6"/> <input type="button" value="List"/> <input type="button" value="Image"/> <input type="button" value="Link"/> <input type="button" value="Code"/>	
<input type="text"/>	
<input type="button" value="Submit"/>	

**CataLoG**

MAIN NAVIGATION

- Dashboard
- Users
- Products
- Categories
- Contacts

Category Add

Form

Title	Parent
<input type="text"/>	<input type="text"/>
Slug	Featured Image
<input type="text"/>	<input type="button" value="Choose File"/> No file chosen
<input type="checkbox"/> Is Active	
<b>Description</b>	
<input type="button" value="Normal text"/> <input type="button" value="Bold"/> <input type="button" value="Italic"/> <input type="button" value="Underline"/> <input type="button" value="H1"/> <input type="button" value="H2"/> <input type="button" value="H3"/> <input type="button" value="H4"/> <input type="button" value="H5"/> <input type="button" value="H6"/> <input type="button" value="List"/> <input type="button" value="Image"/> <input type="button" value="Link"/> <input type="button" value="Code"/>	
<input type="text"/>	
<input type="button" value="Submit"/>	

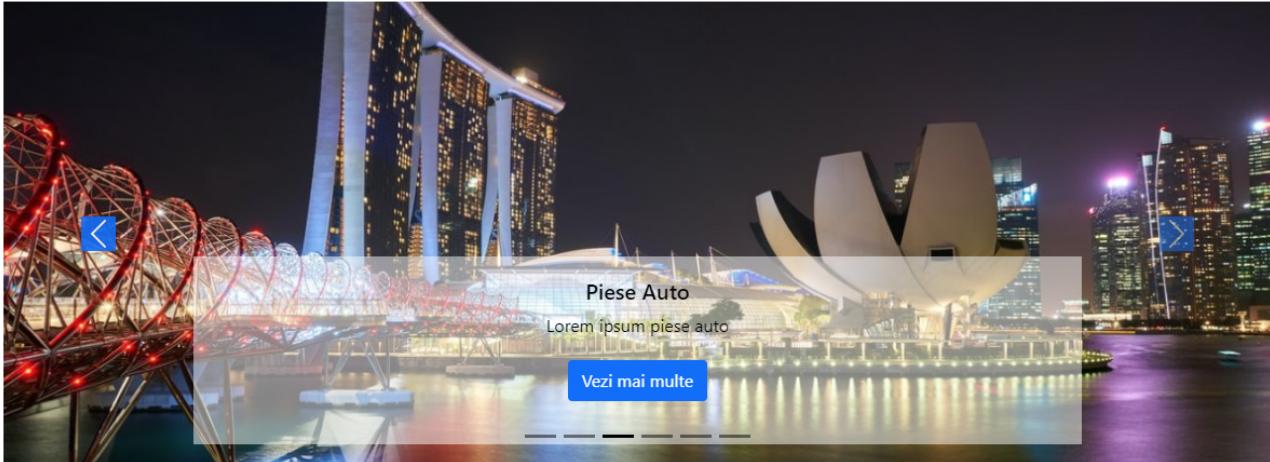
# Front End

CataLoG

Acasă Categori Contact

Caută

Caută



## Categorii



### Materiale de construcții

Lorem ipsum materiale de construcții

[Vezi mai multe](#)



### tesssst cat

tesssst cat Description

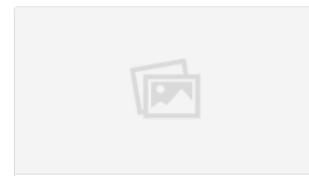
[Vezi mai multe](#)



### Piese Auto

Lorem ipsum piese auto

[Vezi mai multe](#)



### cutie viteze

Lorem ipsum cutie viteze

[Vezi mai multe](#)

CataLoG

Acasă Categori Contact

Caută

Caută

[Acasă / Categori](#)



### Materiale de construcții

Lorem ipsum materiale de construcții

[Vezi mai multe](#)



### Piese Auto

Lorem ipsum piese auto

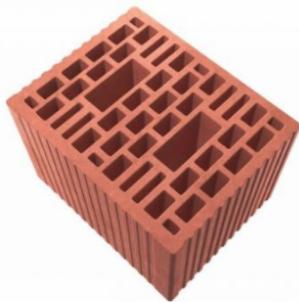
[Vezi mai multe](#)



### test

tessst

[Vezi mai multe](#)

[Acasă](#) / [Categorii](#) / [Materiale De Construcții](#) / [Cărămidă](#)

## Cărămidă

Pret: 2022 LEI

Lorem ipsum Cărămidă



© CataLoG 2021

## Contact

[Acasă](#) / [Contact](#)

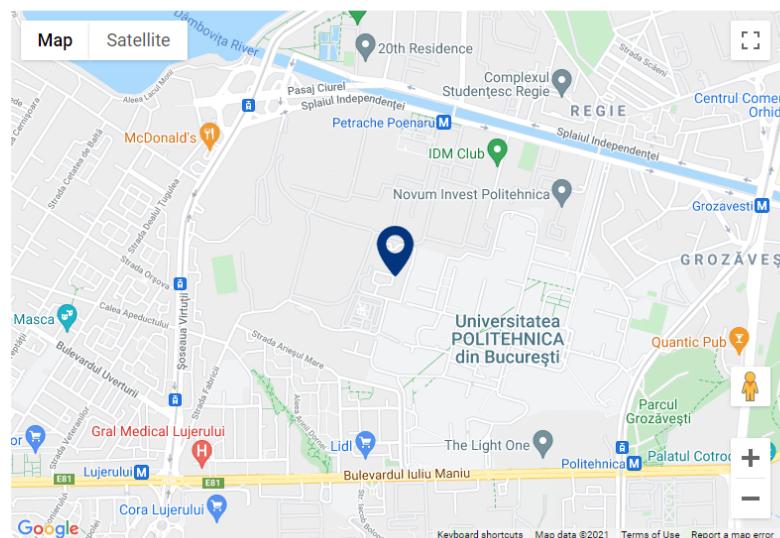
Nume

Telefon

Email

Mesaj

**Trimite**



# Employee Management

## 1. Overview

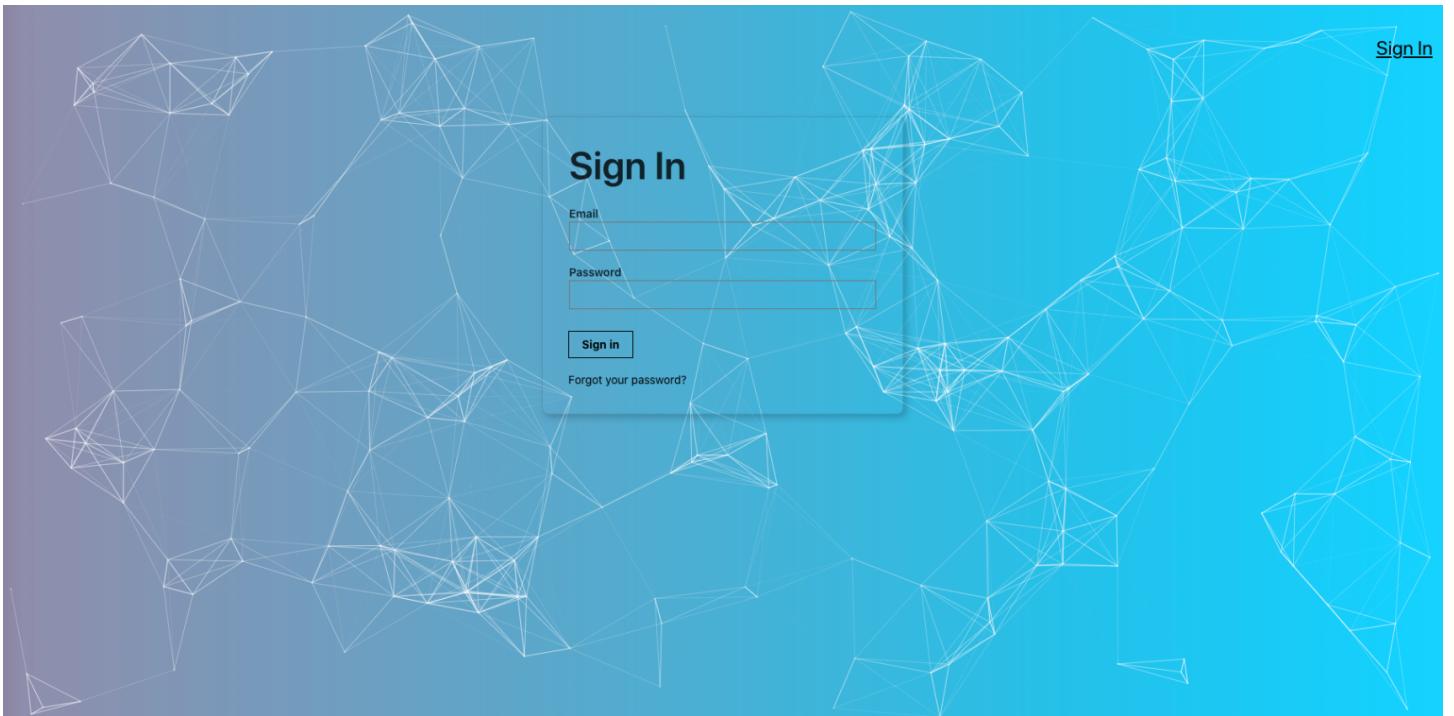
Employee Management is a web application which facilitates employee management from a company or department. It's build having in mind scalability and in the first phase it was developed to be used by Manager or Hr team. Access to the application is restricted to all personal (in first phase), the access is only for people that manage teams.

## 2. Components

Employee Management is composed of two separate applications which can be deployed, developed and integrated independently.

- BackEnd: Is built using NodeJS and ExpressJS which communicate with a Mongo Database using mongoose. All data is stored in mongo cloud. The decision behind choosing a NoSQL DB was that instead of having dependencies between tables for this particular application the data is better stored in a cluster and each employee has the data in one place. It's easier to manage and lowers the risk of corrupting data.
- FrontEnd: Is built on the latest version of React with tachyons, tilt and particles-js which allowed a better customization of the app. Once logged in the user has access to the interface of the application:
  - All employees of the department/company will be listed alongside their data
  - Search is responsive meaning that tipping / erasing a single character will only display employees matching the criteria
  - Employee display area is contained in a "box" and scrollable, this was set like this to not have to have a very big height on the page if there are many employees
  - Add / Remove employees can be done using the menus associated by inputting valid data in the fields

## Login



## Overview

The overview screen has a background with a network of white lines on a blue gradient. On the left side, there is a blue square containing a white icon labeled "LOGO". At the top right, there are three buttons: "Add", "Remove", and "Sign Out". Below these buttons is a search bar with the placeholder text "Search employee". The main area displays a grid of employee profiles. Each profile consists of a small circular icon with a person's face, the employee's name, and their email address. The names and emails are as follows:

Name	Email
Vieriu	asdikasd@test.com
xxxxx asdasd	asdikasd@test.com
xxxxx asdasd	asdikasd@test.com
xxxxx asdasd	asdikasd@test.com
Random	this.state.ema
test	test@account.com
test	test@account.com
test acc	sads@sads.com
test	asdas@c.com
te	sadsa@sadascc.com
terer	test@account.com
test	asdasd@.com
asdasd	alex.vieriu@icloud.com
test	tesad@sas.com
argsssss	alex.vieriu@icloud.com
Ion Popescu	ion.popescu@gmail.com
Andrei Ion	andrei.ion@gmai.com

## Search

The screenshot shows a search interface with a blue header bar containing the word "LOGO" and three navigation links: "Add", "Remove", and "Sign Out". Below the header is a search input field with the placeholder "te" and a close button "x". The main area displays a grid of user profiles. Each profile card contains a small user icon, the name, and the email address. The names and emails listed are:

- test (test@account.com)
- test (test@account.com)
- test acc (sads@sads.com)
- test (asdas@c.com)
- te (sadsa@sadascc.com)
- terer (test@account.com)
- test (asdasd@.com)
- test (tesad@sas.com)

## Add new

The screenshot shows an "Adauga Angajat" (Add Employee) form. The form fields are as follows:

- Nume: Test Account
- Email: test@account.com
- CNP: 91210094940
- Functie: Programator
- Telefon: 0213123123
- Adresa: Bv. Unirii 72

At the bottom of the form is a "Adauga" (Add) button.

## **View added**

