Project plan

ROOSH CHALLENGE 5

ROOSH.

Date	**:**	**10/11/2023**
Version	**:**	**1.1**
Status	**:**	**Open**
Author	**:**	**Group 1**

Table of contents

1. Version management

Version	Date	Author	Changes made
1.0	09-11-2023	Group 1	First setup
1.1	16-11-2023	Group 1	We made changes to: ➢ We added a project definition. ➢ We added made the assignment definition clearer. ➢ We clarified the scope of the project. ➢ We added the conditions of the project. ➢ We added the client questions. ➢ We updated our time plan. ➢ We updated our risks assessment.

2. Project Assignment

2.1 Context

ROOSH is a rapidly growing PaaS company that provides its users with a trustworthy way of reserving parking at the airport. With this growth in the business section also comes a growing demand of the infrastructure that is used for their platform and website. Roosh currently stands at a customer count of 500.000 across 14 counties.

2.2 Problem definition

Its problem is that the user counts differ per day and time. To prevent some parts of the system from slowing down or crashing the number of nodes needs to be increased to handle the load. Hosting is done in the cloud, so ROOSH pays for use. If the system always runs at max capacity, it is really expensive. To summarize, the problem is that the users go up at irregular times so the system would need to run at max capacity to ensure it runs as it is supposed to.

2.3 Goal of the project

The goal of the project is to create a prototype in which an autoscaling solution is tested using Kubernetes, in order to solve the problem of a steadily increasing traffic demand.

2.4 The assignment

The assignment is to create a prototype for auto-scaling in Kubernetes. When the CPU utilization and memory reaches a certain threshold based on the need of the application usually this is around 70-80%, it must be able to automatically create new pods and scale down when the CPU utilization and memory usage drops below 20%. For the prototype we are allowed to use a premade application with a frontend and backend for testing purposes. When there is enough time remaining in the project and our prototype has been accepted, we must automate this process.

2.5 Scope

The scope for this project is to create a prototype of a Kubernetes cluster with autoscaling. This is the core of the project and is our first priority. As you can see in the table below there are two more entries. These are optional features that we will try to implement if time if available.

ROOSH	M	S	C	W
Creating a prototype for auto-scaling in Kubernetes.	X			
Provide a recommendation about possible load testing tools	X			
We could automate the auto-scaling process using Terraform & Ansible.		X		
Creating a proper monitoring solution for the prototype E.G. Prometheus & Grafana.			X	

2	1	1	
50%	25%	25%	0%

2.6 Conditions

- We should use the Elastic Kubernetes Service in AWS.
- We should use a simple front-end and back-end application, this can be our project website.

2.7 Finished products.

At the end of the project, we will be delivering the following products:

- Project plan
- Design document
- Research document
- Test report
- Working prototype

2.8 Research questions

Research Questions:

1. **How to design an autoscaling Kubernetes cluster in AWS?**

    1.1 What are the key considerations for designing a scalable architecture within a Kubernetes cluster?

    1.2 How can we optimize resource allocation within the cluster to accommodate varying traffic loads efficiently?

    1.3 What are the best practices for configuring horizontal pod autoscaling (HPA) and cluster auto-scaler in Kubernetes?

2. **How to properly simulate traffic loads to test the cluster?**

    2.1 What tools can be employed to simulate realistic traffic patterns for accurate testing?

    2.2 How can we ensure that the simulated traffic reflects real-world scenarios, including peak usage and sudden spikes?

    2.3 Are there any standards for testing Kubernetes clusters under different traffic conditions?

3. **How to automate deploying the cluster using Terraform and/or Ansible in AWS?**

    3.1 What are the essential steps and best practices for automating the deployment and scaling processes using Terraform —and/or Ansible?

    3.2 How can automation scripts be adapted to handle updates, rollbacks, and other maintenance tasks in a Kubernetes environment?

        3.2.1 Which measures can be taken to ensure reliability and security for automating the auto-scaling process?

4. Monitoring solution (optional)

    4.1 What metrics and KPIs should be used for effective up and down scaling?

    4.2 How can we integrate Prometheus and Grafana into the Kubernetes environment?

2.9 Client Questions

1. **What is the exact problem ROOSH is facing?**

2. **How do you envision the autoscaling feature to work?**
   2.1 Are there any specific thresholds for scaling up and down that you have in mind?
   2.2 What type of application do you want us to you to create this auto scaling prototype?
   2.3 How do you want us to test the load balancer, do we have to use a specific tool?

3. **How do you envision your prototype to look like at the event of the project?**
   3.1 What parts would you like to see automated?
   3.2 What are the conditions for this prototype?

4. **What would you like to for document e.g., design document, testing rapport etc.?**

5. **What would you like to see for the first demonstration?**

6. **When are you available for meetings, questions etc.?**

7. **Is the intern dependent on our work to progress?**

8. **How can we contact the intern?**

3. Approach and Planning

3.1 Approach

For this project the team has chosen the scrum method of managing the project. We will have four sprints with an MVP on week 2, 4, 6 and for the final presentation week 8.

3.2 Breakdown of the project

Starting the first sprint we will begin with doing research on how to properly design the Kubernetes cluster with autoscaling. In parallel with this research, we will also start creating the cluster. By the end of sprint 1 the research should be complete and the start of the creation of the cluster should have been started. For the second sprint we will work entirely on building the infrastructure. When this is done by sprint 2, we can move on to testing. In sprint 3 the design document will be made along with testing of the cluster. For the last sprint we will work on automation, cleaning up unfinished work and preparing the final MVP.

3.3 Time plan

Below you can find our time plan with phasing and deliverables for each phase.

Phasing	Effort and end product	Start	Ready
0 Introduction with the project	First version project plan	9/11/2023	16/11/2023
1 First deployment & monitoring	Demo of first deployment	16/11/2023	30/11/2023
2 Auto-scaling	Demo of second deployment	30/11/2023	14/12/2023
3 Auto-scaling & stress testing	Proof of concept	14/12/2023	25/01/2024

4. Project Organization

4.1 Team members

Our team consists of four members, we have divided roles as seen in the table below. As a team we adopt our accustomed way of working into this project. Meaning that we work closely together as a team but also with the client and other stakeholders involved.

5. Risks

5.1 Risks and fall-back activities

Given the nature of this project our list of risks will be quite short. Most of these risks regard the team itself and lie in the trend of not being able to put time in to work.

Risk	Prevention activities included in plan	Back-up Activities
1 One or more members fall out due to sickness.	Clear communication and tools for members to work from home if possible	Other members put more time in.
2 The Fontys assignments demands more than expected.	Keep a clear tab on what needs to be done for what project and keep up with planning.	Spend less time on assignments to clear up time.
3 Technical difficulties cause work interruptions	Regular maintenance and updates on project related hardware & software & train team members to troubleshoot common issues faced.	Backups of the project in case issues arise with new versions & system for logging to find the root cause of issues during development.
4 Mikaeil or Intern are not available for questions which cause delays	Proper planning and get as much clear as soon as possible.	We temporarily switch to working on a different part of the project

Risk ID	Risk name	Probability	Impact
1.	Sickness of team members	LOW	LOW
2.	Fontys assignment demand more than expected	LOW	MEDIUM
3.	Technical difficulties	LOW	HIGH
4.	Mikaeil or Intern are not available	LOW	MEDIUM

The mentioned risks are classified by their likelihood and severity of the impact they pose on the infrastructure.