TASK: DEVOPS 01

**Name:** Dimitar Lazarov

**Date Finished:**  26.09.2023

TABLE OF CONTENTS:

[**1. Instructions 2**](#_orveuqykry5j)

[a. Estimations: 2](#_bkaeagzg8syk)

[b. Results: 2](#_nsxyjiqp2jmc)

[**2. Goals and Requirements 2**](#_r2pifugwoxqk)

[**3. Bonus points 3**](#_usygabj03qy7)

[**4. Results 3**](#_tcx91w64fzqz)

[a. Estimations: 3](#_izxep6i0y22j)

[b. Comments: 4](#_qw9m5zt6chlk)

[c. Demo: 4](#_s9hvjyrtz300)

[**Contact Us 4**](#_fzwa6ym6o6iq)

# Instructions

Please read the whole document before starting the task.

## Estimations:

* Please break the task down to smaller **subtasks** and give a rough estimation in the table under the Results section.
* It is completely normal if the estimations have a big difference than the time it took to finish the task.
* Estimations are important, please **log the hours** needed for research/implementation of each subtask in the Results -> Accomplished column.
* Also please **reply to our representative** with the initial estimations before starting, so we know roughly how much time you will need.

## Results:

* After finishing the task, please fill in your Name and Date of finish on top of this document.
* The results must go in the **Results -> Demo** table as well.

# Goals and Requirements

Create a local Minikube cluster and install Istio (with all of its services) on it using Helm. After that pick a demo/example app from the official Helm registry, which must be written in Golang or Python and install that as well. At the end expose all tools that are listening on ports and provide screenshots (ex. logging, APM, service mesh).

You have complete freedom regarding:

* Technical solutions that are not strictly defined in above requirements

# Bonus points

For:

* Having a proper Markdown readme;
* Following the appropriate coding standards and best practices;
* Using the SOLID/KISS/DRY principles;
* Having proper comments;
* Implementation of any type of optimizations;
* Implementation of good security practices;
* Attention to detail;

# Results

## Estimations:

|  |  |  |
| --- | --- | --- |
| **Task Name** | **Estimated Time (h)** | **Accomplished for (h)** |
| Install and configure VM running docker,kubectl and minikube. | 1 | <1 |
| Installing HELM, reading documentation (how it works). | 1 | ~1 |
| Installing ISTIO, reading documentation to understand what it does and how it works. | 2 | 3+ |
| Finding and installing a [Python](https://artifacthub.io/packages/helm/python-web-app/reload-counter) application using HELM from the Artifactory HUB | 30 min | 15min |
|  |  |  |
|  |  |  |

* *Please insert more rows if needed.*

## Comments:

|  |
| --- |
| If you have any comments or something that we need to note while checking your results, please enter them here: |

|  |
| --- |
| The install and initial configuration was done in an hour following the documentation. However, understanding the purpose of istio took a lot of time and still much is unclear.  * The python application port is forwarded and can be accessed with (cluster ip) minikube ip:30009 using nodeport. I think this can be done also using istio (gateway ingress)?       The deployment replication is set to 1 pod and the second pot is the envoy proxy that ISTIO deploys as a sidecar and should be listening to all traffic to the service. |

## Demo:

If the task requires a live demo and/or repository URL, please provide it here:

|  |  |
| --- | --- |
| **Description** | **URL** |
| Local vagrant automation and all scripts included. |  |
|  |  |

* *Please insert more rows if needed.*

# Contact Us

If you are stuck or have any assistance, please contact us at:  
[office@theorigamicorporation.com](mailto:office@theorigamicorporation.com)

Good luck!