# **Description:**

In a typical Kubernetes (k8s) cluster no matter its Red hat OpenShift or any other k8s based cluster, It has lots of YAML files. Every resource , whether is a simple pod, or a deployment,DaemonSet or Statefulset or a Persistent Volume, It has lots of Configurations and Many parameters to check. Only running *“kubectl get” or “kubectl describe”* commands and checking line by line is really a tedious job. Most of the time we may miss the critical info from the jungle of Texts.

In some critical issues, when the Support engineer is checking the user’s cluster, checking these files and reading them line by line is a very tedious and stressful job.

So what is the remedy ? Anyway can we solve this problem ? Or is there any way we can get rid of checking YAML files line by line ?

Well let’s think in another way …

If someone is explaining something to me, and I am just listening to them verbally or just reading some text or theories ,will I understand easier ? NO, Either It will make me feel bored or somehow I will lose interest.

On the other hand someone started explaining to me the same topic or concepts with some nice pictures, or figures diagrammatically. Will I understand better now ?

Yes, indeed… It will be more fun. Anything which can be represented pictorially or visually we get much more fun from it. Anything can be explained far better visually than only by a written text format. No matter if it’s a Presentation or any hard mathematical or scientific problem …

Hence I want to make a program , a kind of tool or method by which we can represent a kubernetes cluster resources into visual diagrams to denote the interrelationships in between them and to break down the Resource entities in smaller pieces to dig down better…

But how ? Do I need to install something on User’s kubernetes cluster ? But maybe I am not authorized to do so. It may create a Security concern Exactly . Indeed, it’s a concern and we don’t want our users to install a new tool on their clusters. It’s a tool a Support engineer will use. Users just must provide the engineer one INPUT YAML File . That’s it.

The tool the engineer should run in his machine to grab an overview about the user’s kubernetes cluster.

So, Finally, The tool to our rescue is : ***MERMAID***

Reference:

* <https://mermaid.js.org/>
* <https://pypi.org/project/mermaid-py/>
* <https://github.com/mermaid-js/mermaid>
* <https://kubernetes.io/docs/contribute/style/diagram-guide/>

***Mermaid is a JavaScript-based diagramming and charting tool that uses Markdown-inspired text definitions and a renderer to create and modify complex diagrams.*** The main purpose of Mermaid is to help documentation catch up with development. It uses a simple, markdown-like syntax that allows users to define diagrams directly within their markdown files. This method is particularly useful for creating diagrams that visually represent Kubernetes concepts, workflows, or architectures.

***Mermaid lets you create diagrams and visualizations using text and cod*e.** Mermaid is the tool that reads and interprets this code to generate the visual diagrams. This approach allows you to create and modify diagrams using text-based syntax, making them easy to version control and share.

Let’s talk about **.MMD** file

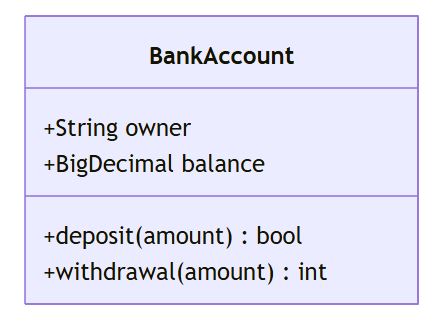
**A .mmd file is a m**er**m**aid **d**iagram FILE. Some people call it the **Mermaid Markdown file** also. But actually Mermaid Markdown means, markdown containing text in markdown syntax extended with the capability to render mermaid diagram

It contains the text-based syntax used to define and create diagrams using the Mermaid diagramming library. Mermaid diagrams are generated from this ***Markdown-inspired syntax*** and can be used to represent various types of diagrams, such as flowcharts, sequence diagrams, and class diagrams.

.MMD or **Mermaid Diagram file i**s a text file where you write instructions for drawing a diagram.

Let’s take an example of a class

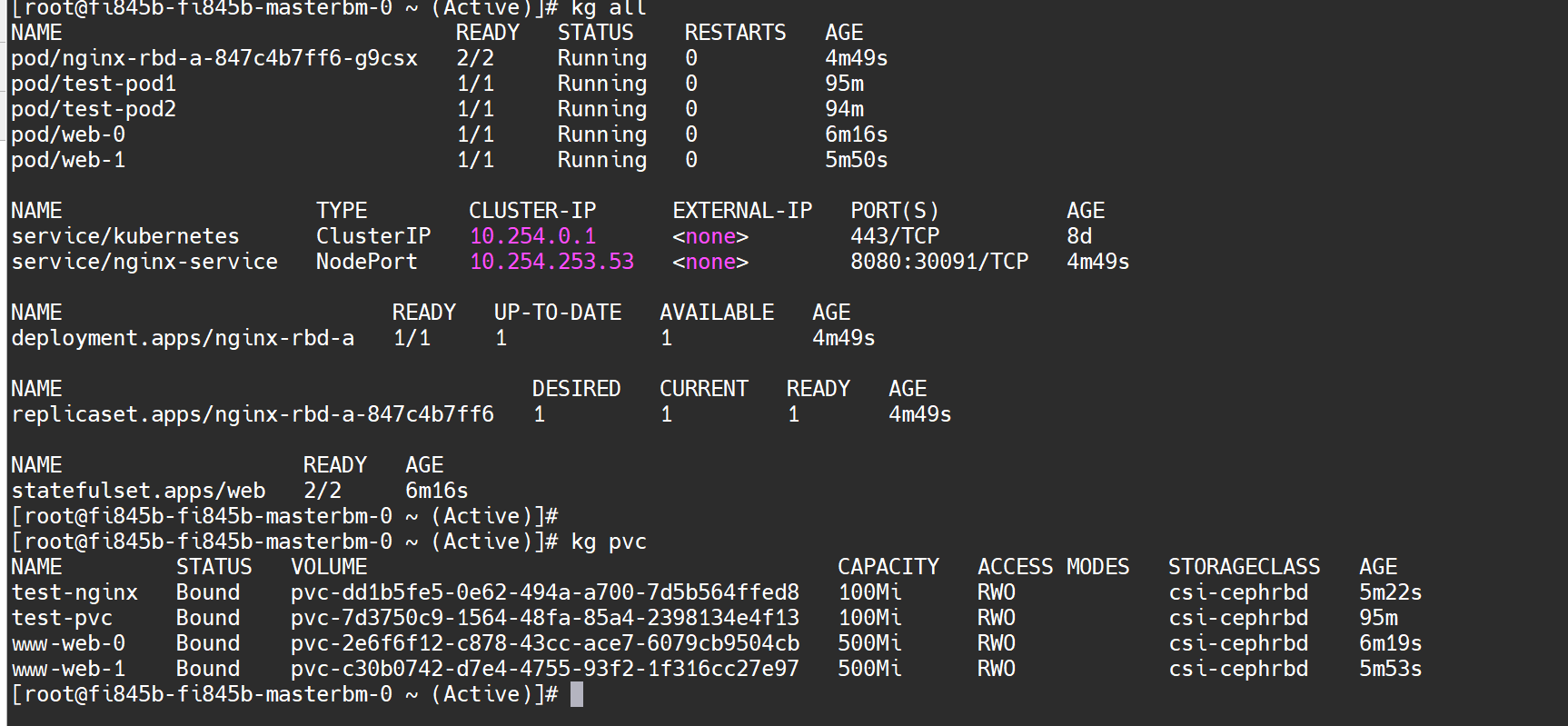
|  |
| --- |
| classDiagram  class BankAccount{  +String owner  +BigDecimal balance  +deposit(amount) bool  +withdrawal(amount) int  } |



# **What we want to achieve here**

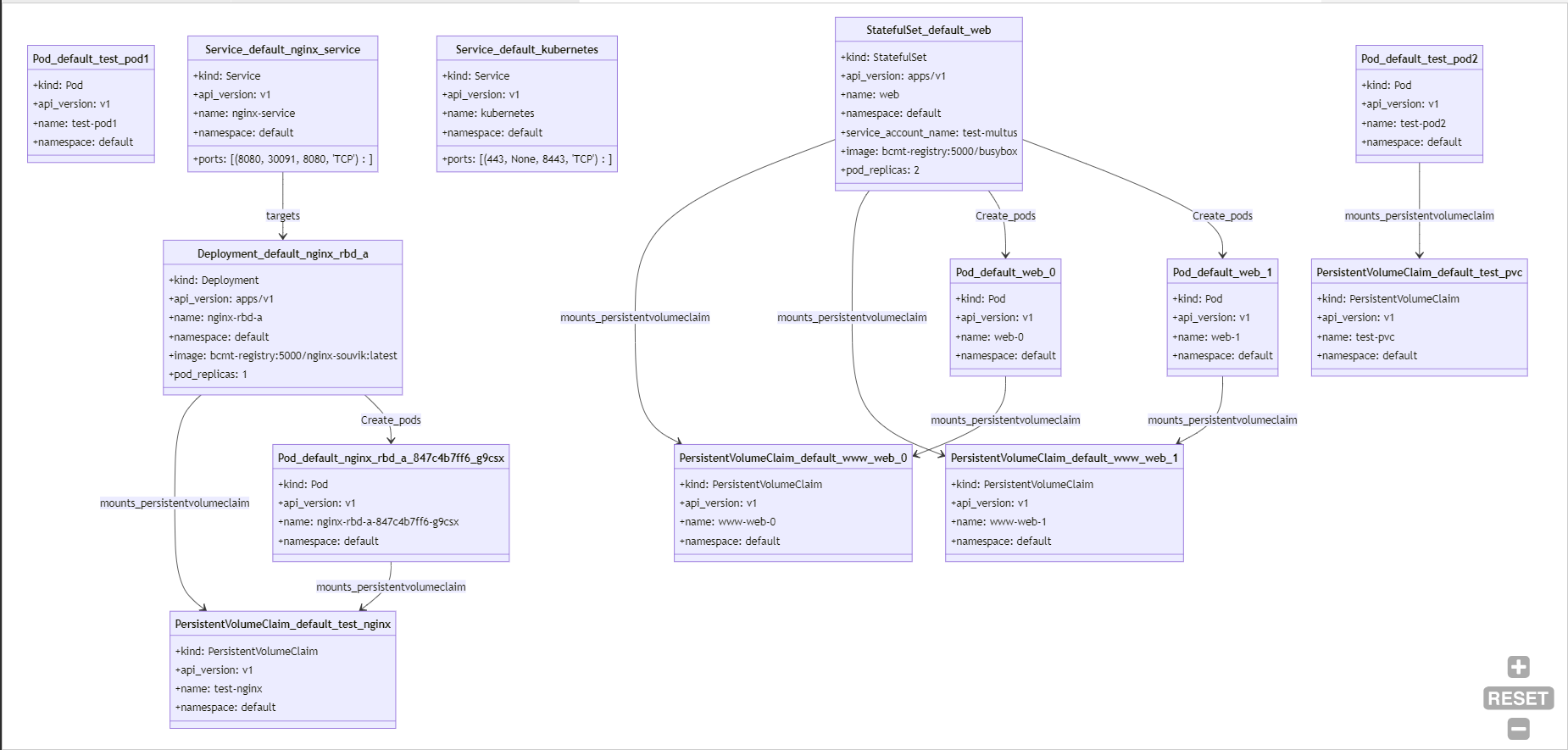
What we want to achieve is this ..

Suppose we have some workload pods running in one namespace.



Now these above pods and mount volumes we have . We want to convert this resource in a class diagram format using MERMAID

Please see below.



So here if we see above, My kubernetes resources are shown in diagram format. A daemonset/ deployment/statefullset creates one or many pods.

A deployment, Daemonset and Statefullset has mount volumes. Same PODS too have mount volumes attached.

Let’s see the USER INPUT YAML file.

|  |
| --- |
| **kubectl get all,pvc -n default -o yaml > FILE\_NAME.yaml**  **kubectl get all,pvc,secret,cm -n kube-system -o yaml > FILE\_NAME.yaml** |

Provide the **FILE\_NAME.yaml to the Support Engineer**

# 

# **How to Run the tool**

Our python file name is : **mermaid-k8s-all.py**

We have to run this file and need to send the YAML input file name as a command line argument.

|  |
| --- |
| **Python3 mermaid-k8s-all.py NAME\_OF\_THE\_YAML\_FILE\_INPUT\_FROM\_USER\_CLUSTER**  **python3 mermaid-k8s-all.py all.yaml** |

Once the python file runs successfully, It will generate an output .MMD file: Mermaid\_class\_diagram\_output.mmd

Now this MMD file can open in any editor (where Mermaid plugin installed) and can view the resources like above.