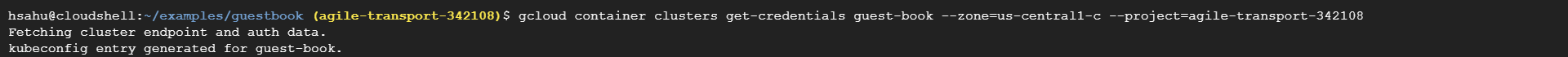
1. Project: Build a Scalable Guest Book Application We will create a Redis Database storage of your guestbook, start the guestbook web servers. The Redis workers, is a replicated application managed by a deployment. This Guest Book uses a PHP frontend. The Guest Book is configured to talk to either the Redis worker or master services, depending on whether the request is a read or a write. It exposes a simple JSON interface and serves a user experience based on jQuery and Ajax.

So we have a GitHub repo where we have all the configuration files where we can get started i.e. “yaml” files. The link of the repo is <https://github.com/kubernetes/examples>. Now let’s get started with the creating cluster using cloud shell and clone the repo containing the guestbook files.

Text

Description automatically generated Text

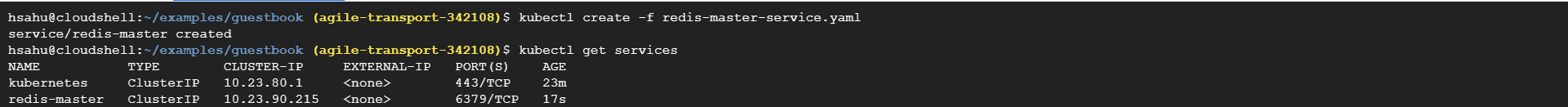
Description automatically generated

Now we need to work on the kubernetes so we need to get authorized so again we need to use cloud shell in order to get authorized

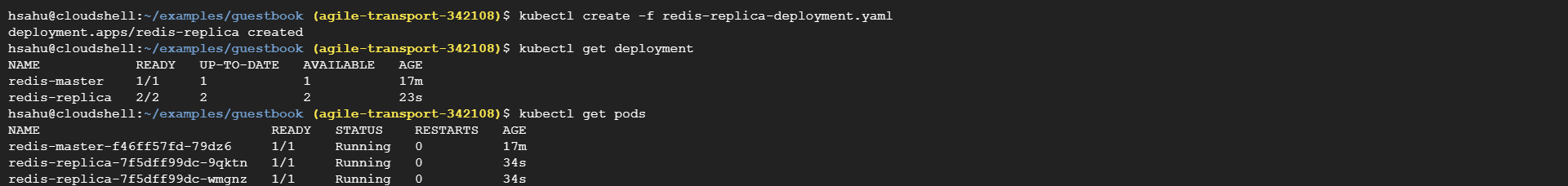
Now we have authority to use kubectl or kubernetes engine tasks. So, next we need to deploy a controler and we have config file for that so first let’s read the configuraion and then deploy a controlerText

Description automatically generated

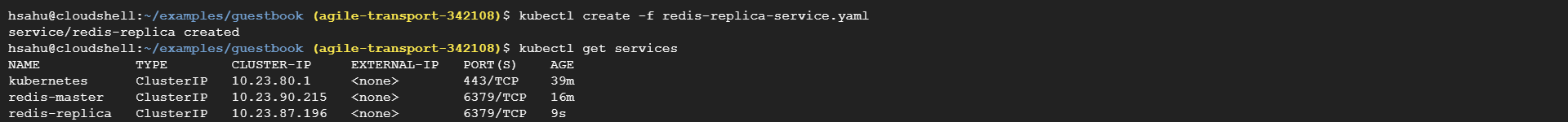
Now we need to create a Service in order to handel the requests and get the external IP. Lets view the configuration and create a service. Graphical user interface, text

Description automatically generated

As we know radius master is a single pod who is handling the requests so we need to add some workers node to it so it can handle the more requests and increses performance. Lets see the configuraion and deploy a worker nodeText

Description automatically generated

Now we need to create services for the worker node let’s see the configuration of workes node service an deploy it Text

Description automatically generated

Set up the guestbook web frontend. Now that you have the Redis storage of your guestbook up and running, start the guestbook web servers. Like the Redis workers, this is a replicated application managed by a deployment. The guest application use a PHP frontend. It is configured to talk to either the Redis worker or master services, depending on whether the request is a read or a write. It exposes a simple JSON interface and serves a user experience based on jQuery and Ajax. Text

Description automatically generatedText

Description automatically generated

Now need to deploy service for frontend and expose the service to external world so it can be accessableText

Description automatically generatedText

Description automatically generated

Now we can varify the deployment by vising the external IP or by curling the requestsText

Description automatically generatedGraphical user interface, application, website

Description automatically generated