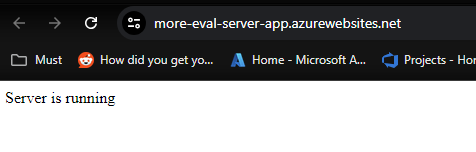
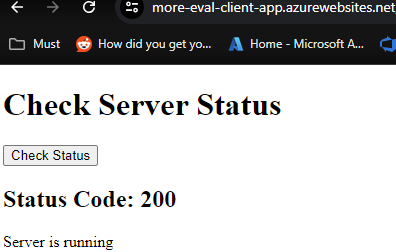
More.com assessment Project:

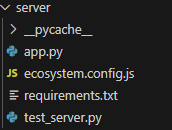
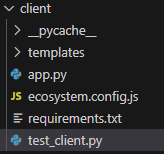
* App: Python/Flask
* Infrstructure: Azure/Local with IaaS Terraform
* CI/CD: Azure DevOps
* Monitoring/Alerting: Azure Monitoring and Dashboards

1. The application

The application consists of 2 separate simple projects/apps. The concept is a client/server stack.

Server’s job is just to run and return it’s status  
https://more-eval-server-app.azurewebsites.net/  
  
  
  
  
Client’s job is to ask server about it’s status and post it.  
https://more-eval-client-app.azurewebsites.net/  
  


Both apps are very similar and their purpose is to have something to set the CI/CD pipelines on.

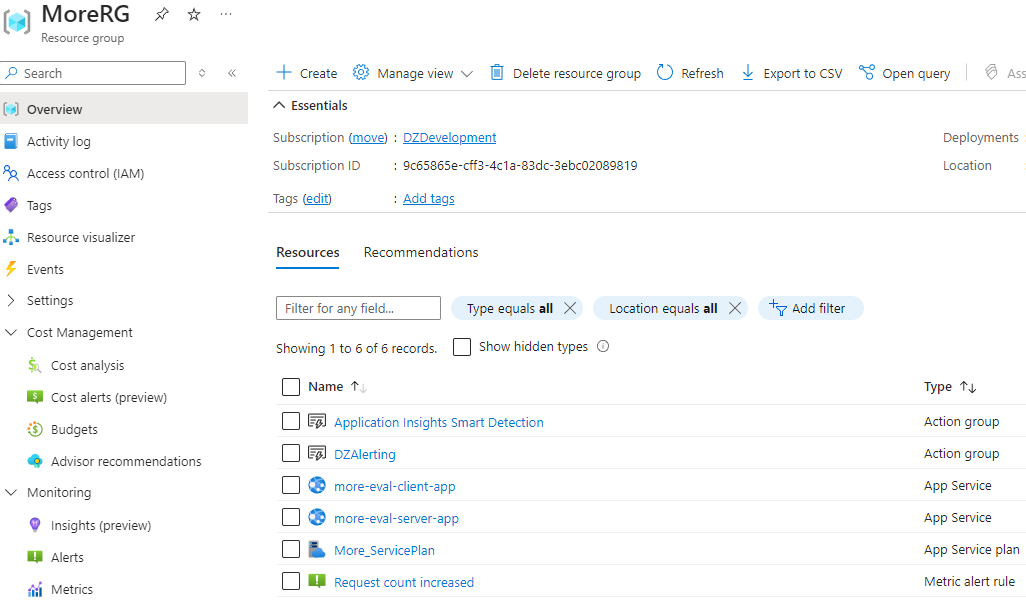
  
  
  
I am using Python and Flask to set the apps.

* In the app’s directories you will also find the **test\_client.py** and **test\_server.py**, these are the tests.
* **ecosystem.config.js** will be used with pm2 tools to manage local deployments of the application.

2. Infrstructure

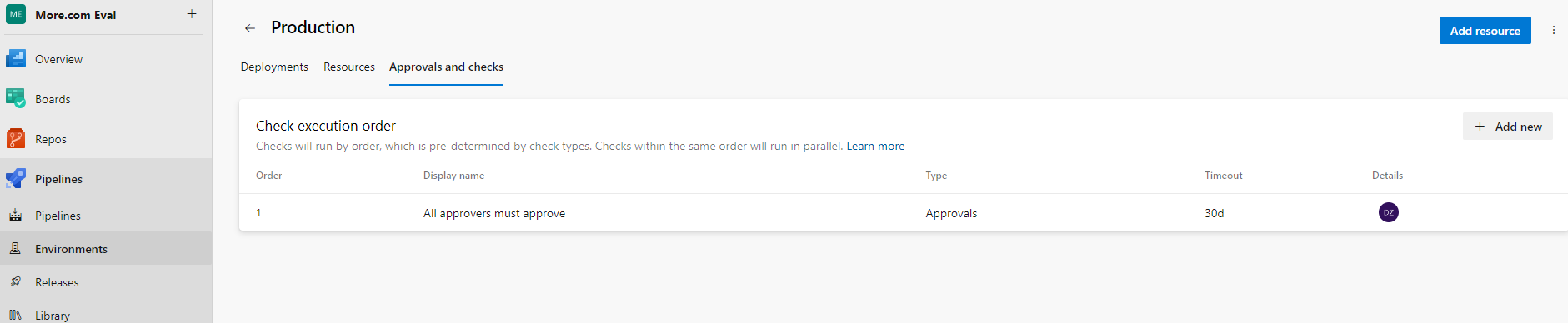
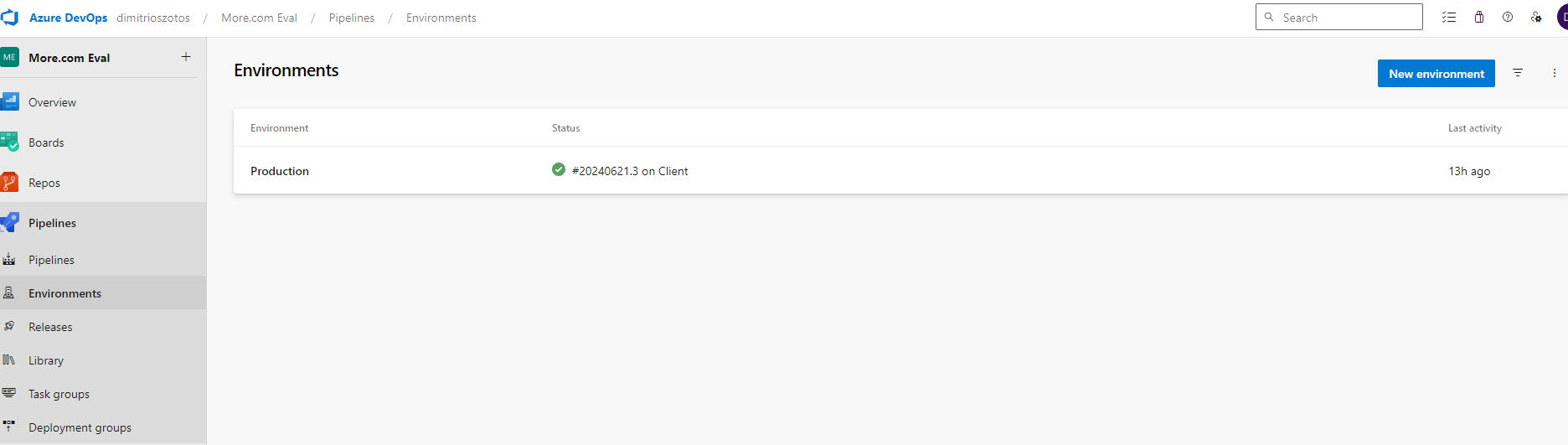
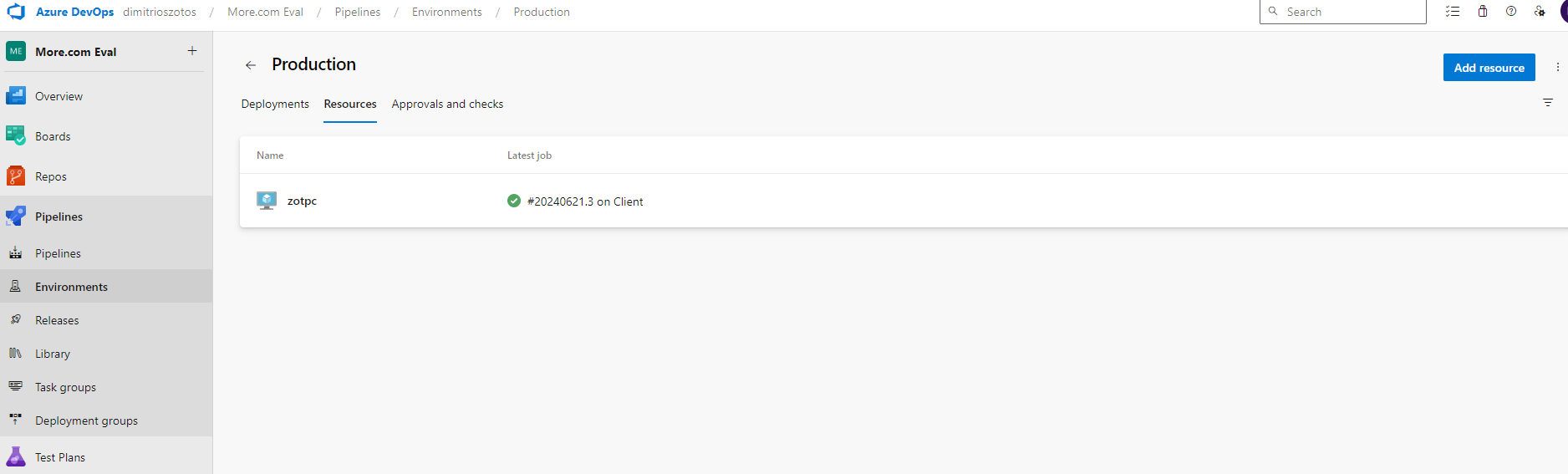
I am using Azure for the deployments of:

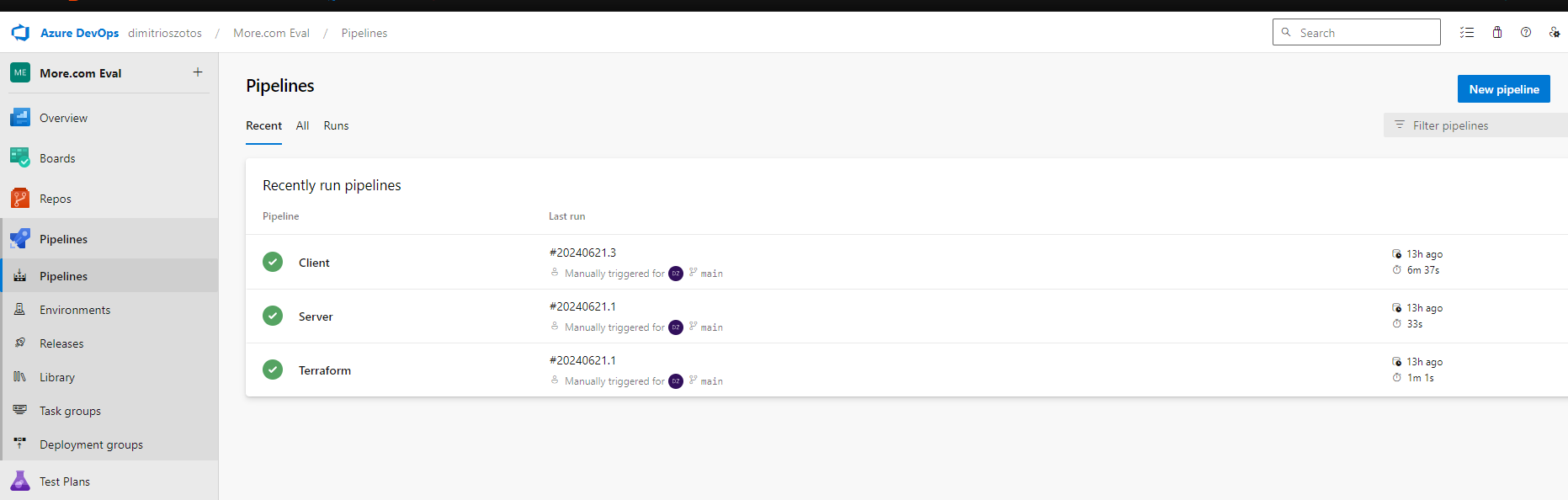
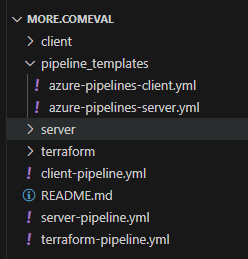
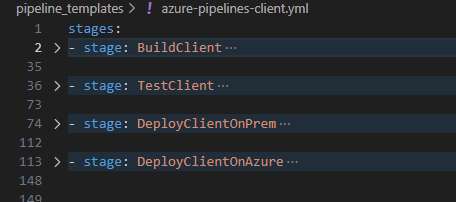
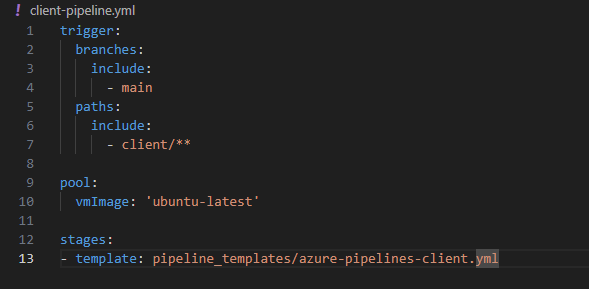
* 2 Resource Groups. One will hold everything related to the Project (MoreRG) and the other (DZDevelopment) will hold a storage account where we store our Terraform State files
* 1 Service plan (More\_servicePlan) that host the App Services
* 2 App Services (more-eval-client-app and more-eval-server-app)
* Metric alert rules and actions (email when metric reaches a threshold)
* 2 Dasboards to monitor our apps



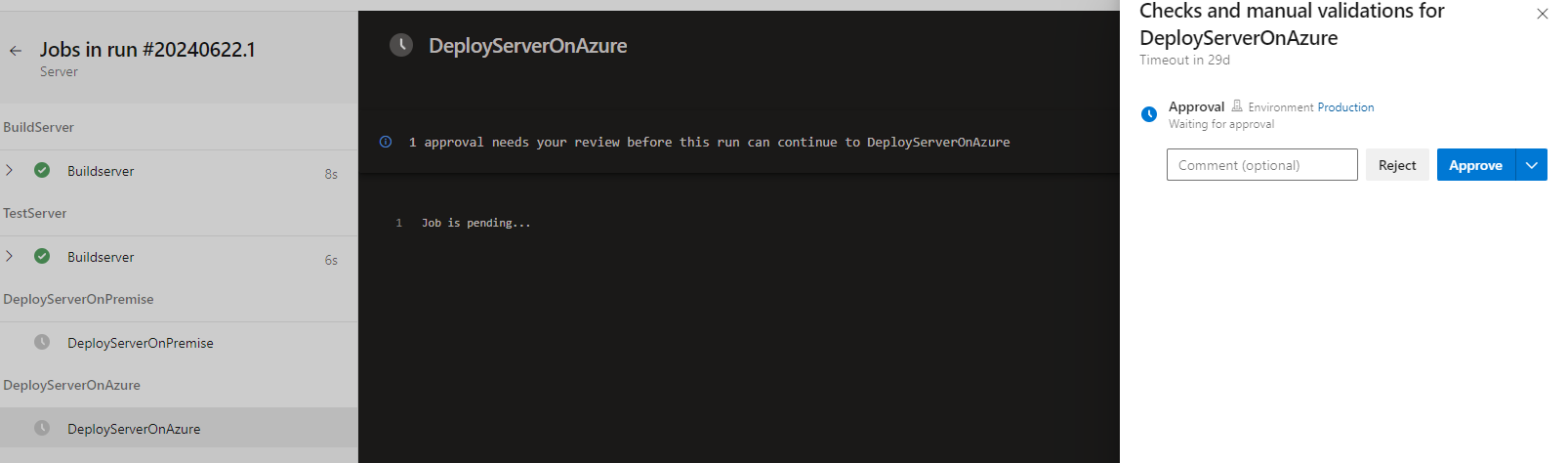
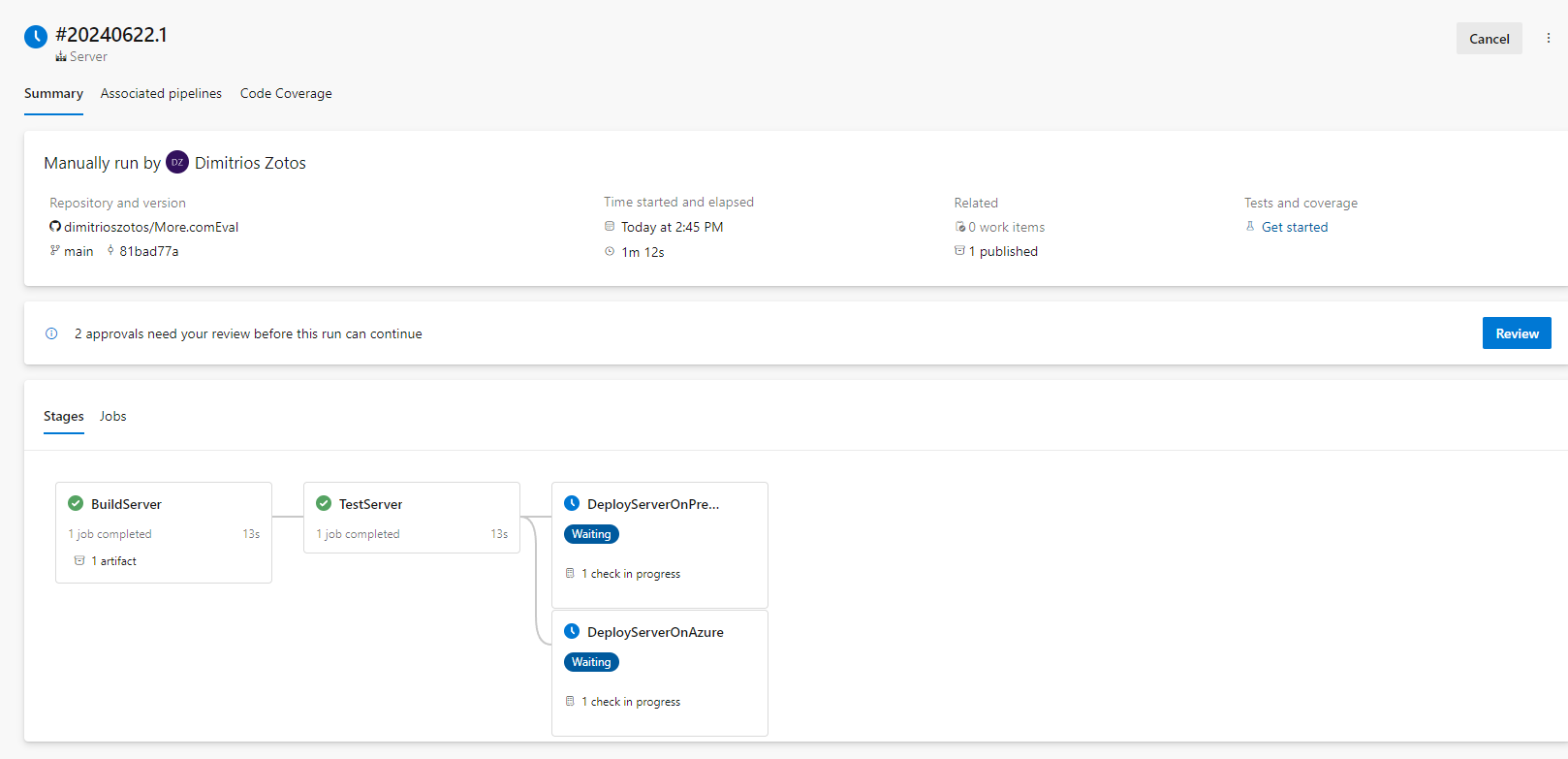
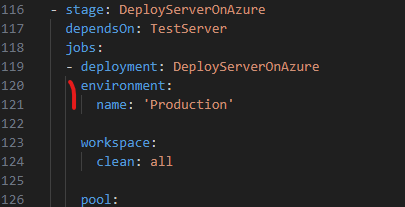
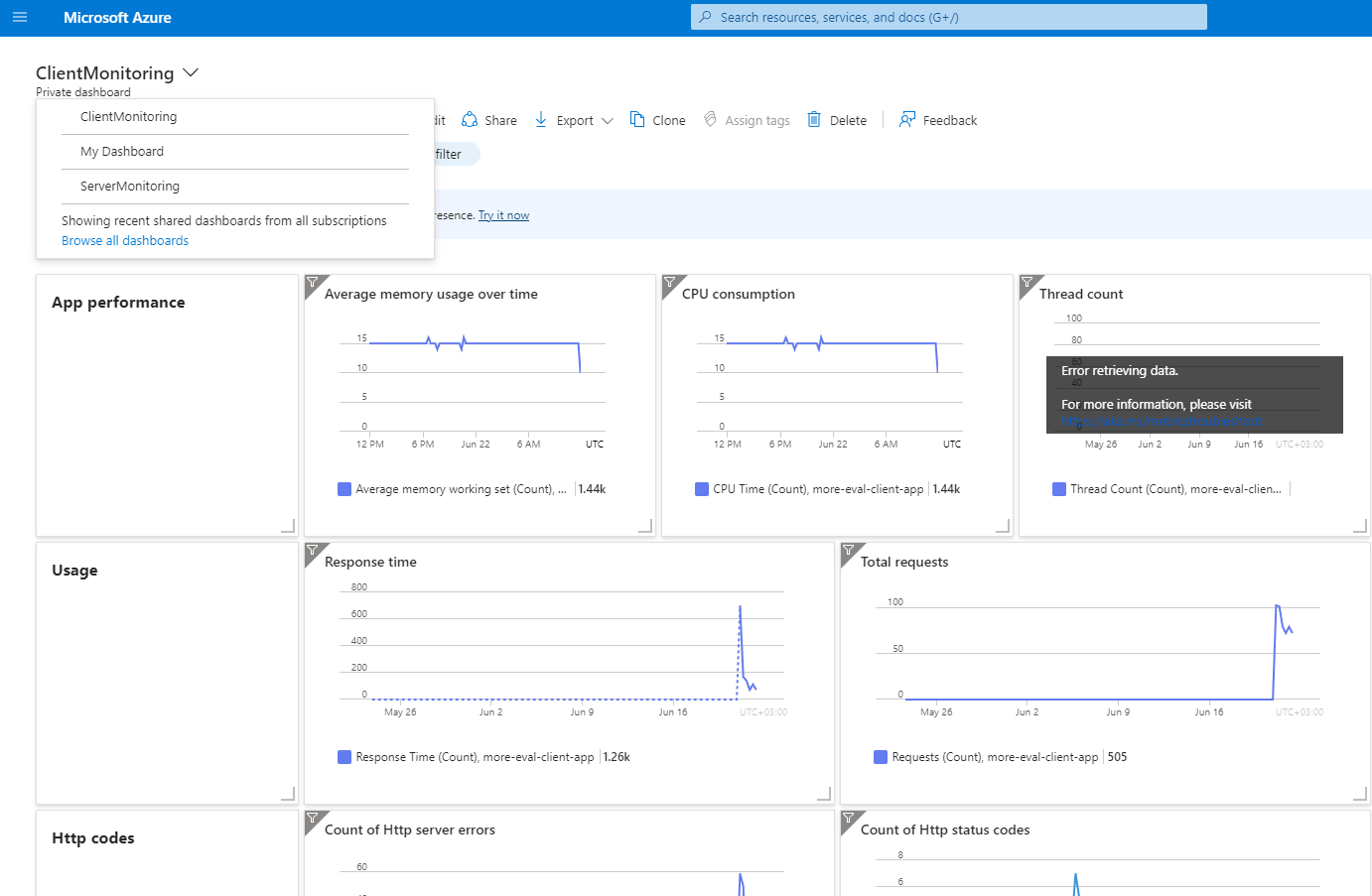
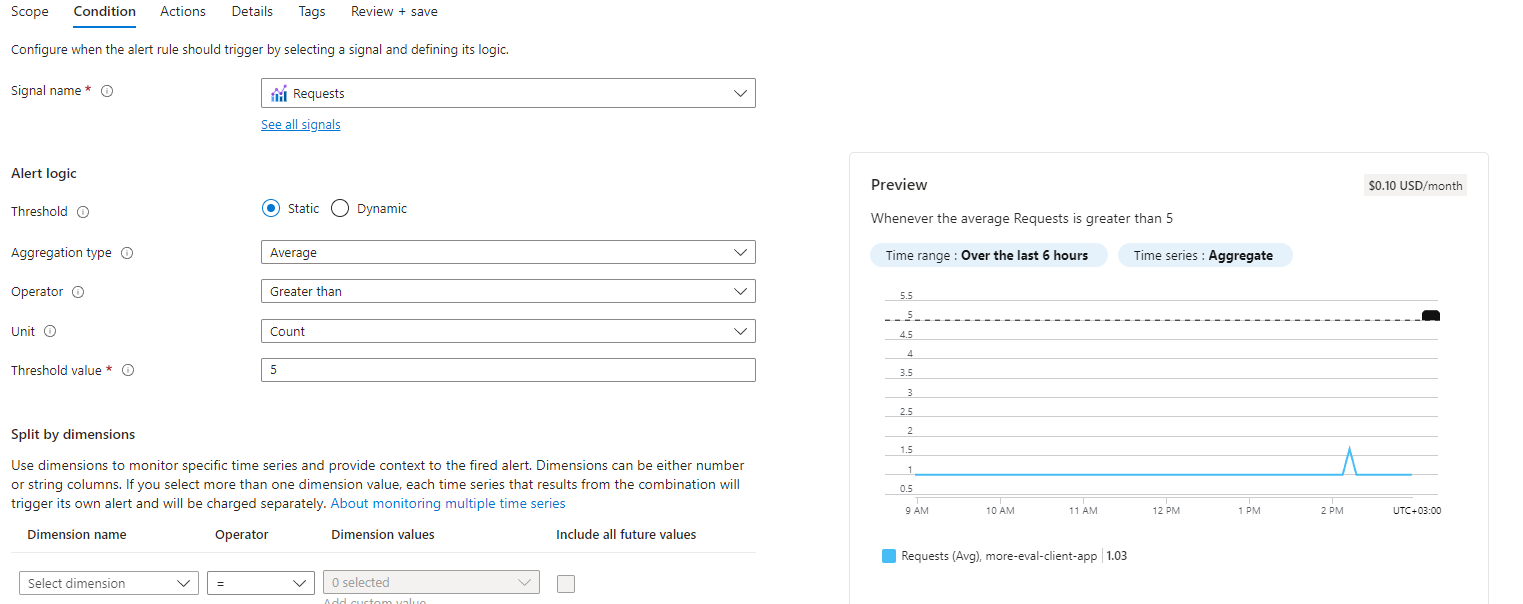
3. CI/CD Azure DevOps

For the CI/CD part I am using Azure DevOps.  
  
I set the “More.com Eval” project under my space and then with the set-up.  
  
In order we have the “Approval” feature on our deployments we need to set up an Environment

  
Also here, I set my own hosted server as a target for 1 of the 2 Production deployments:  
  
\* The environment name ”Production” is going to be used in the CI/CD pipelines to set the Approval before deployment.

After that we are ready to proceed setting our pipelines.  
I import the pipelines I already wrote on my project:  
  
As you can see in the repo:  
  
we have 3 pipelines and 2 templates.  
A pipeline for each of Client/Server and one for Terraform so we deploy our infrastructure on Azure.  
The 2 Client/Server pipelines will call the corresponding template pipeline to run.  
  
I could split the yml pipeline files on more templates for each stage for better visibility and manageability but for this scale it isn’t required.  
  
Terrafom pipeline will deploy the infrastructure described above on Azure.  
  
  
  
On triggers:

I have set a trigger to both **client** and **server** pipelines. They will trigger when any changes are pushed into the “Main” on the client directory for **Client app** and the same but for server directory in our repo for **Server App.**

Running the Pipelines:  
Manually or by trigger the pipelines will go through the Build Test and Deploy stages.  
Deployment has 2 separate tasks, 1 will deploy on my local Server and the other one our Azure Cloud App Service.  
Both require approval.  
\*Approval set when we added our Environment ”Production” and called in the pipelines from here  
  
4. Monitoring/Alerting: Azure Monitoring and Dashboards  
  
To monitor the deployment I am using the Azure Dashboards.  
I created a Dashboard for each of our app.  
  
Alerts also can be set here.  
  
 For example: When the average count of requests on the Client is more than 5:  
It will trigger an action that will send me an email.