Documentation:

Steps included in the process:

Step - 1: Created an AWS Elastic Search cluster

Step - 2: Imported data into the cluster

Step - 3: Wrote queries to search data

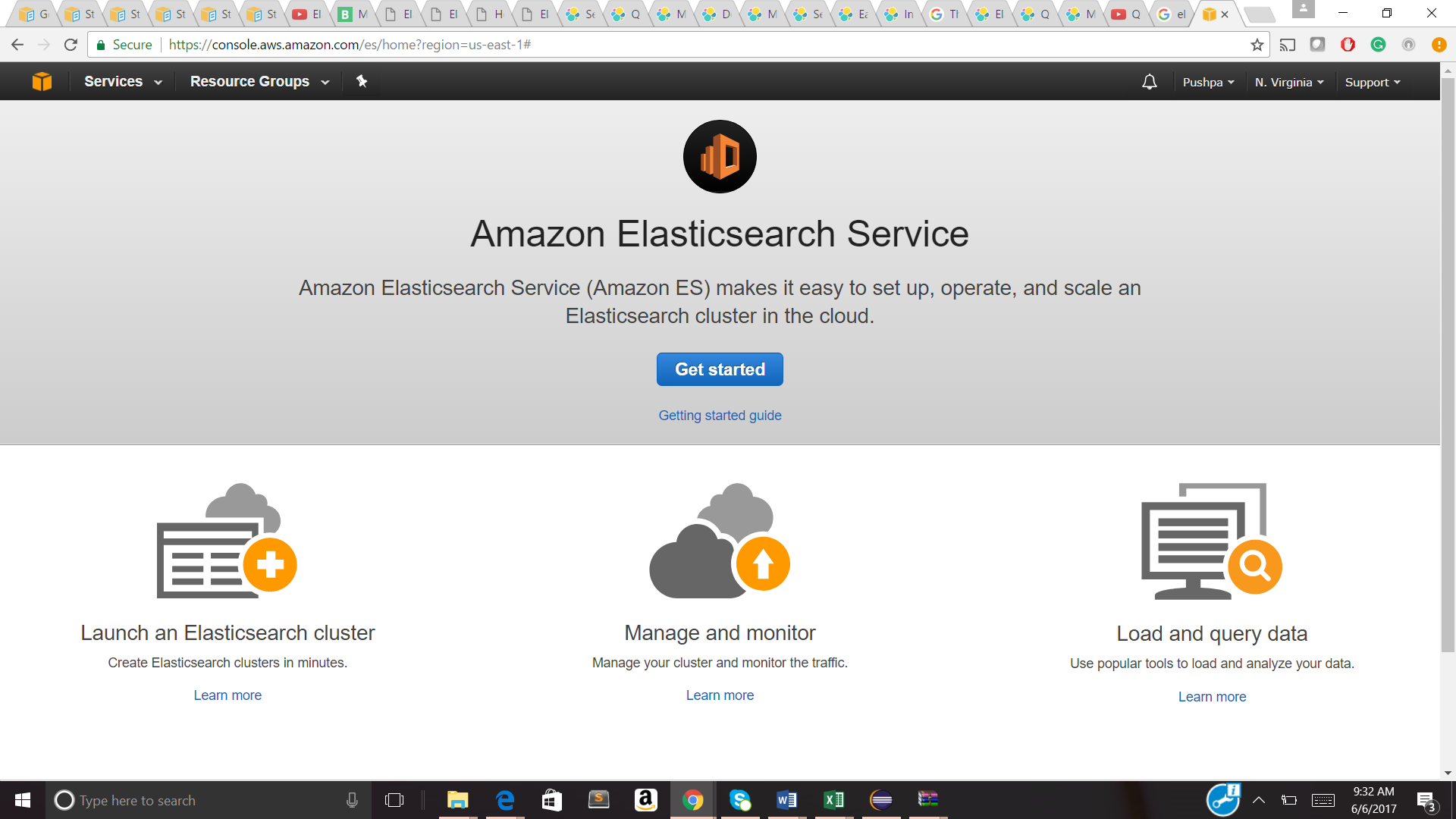
Step - 4: Created a AWS Lambda Microservice which triggers by AWS API Gateway

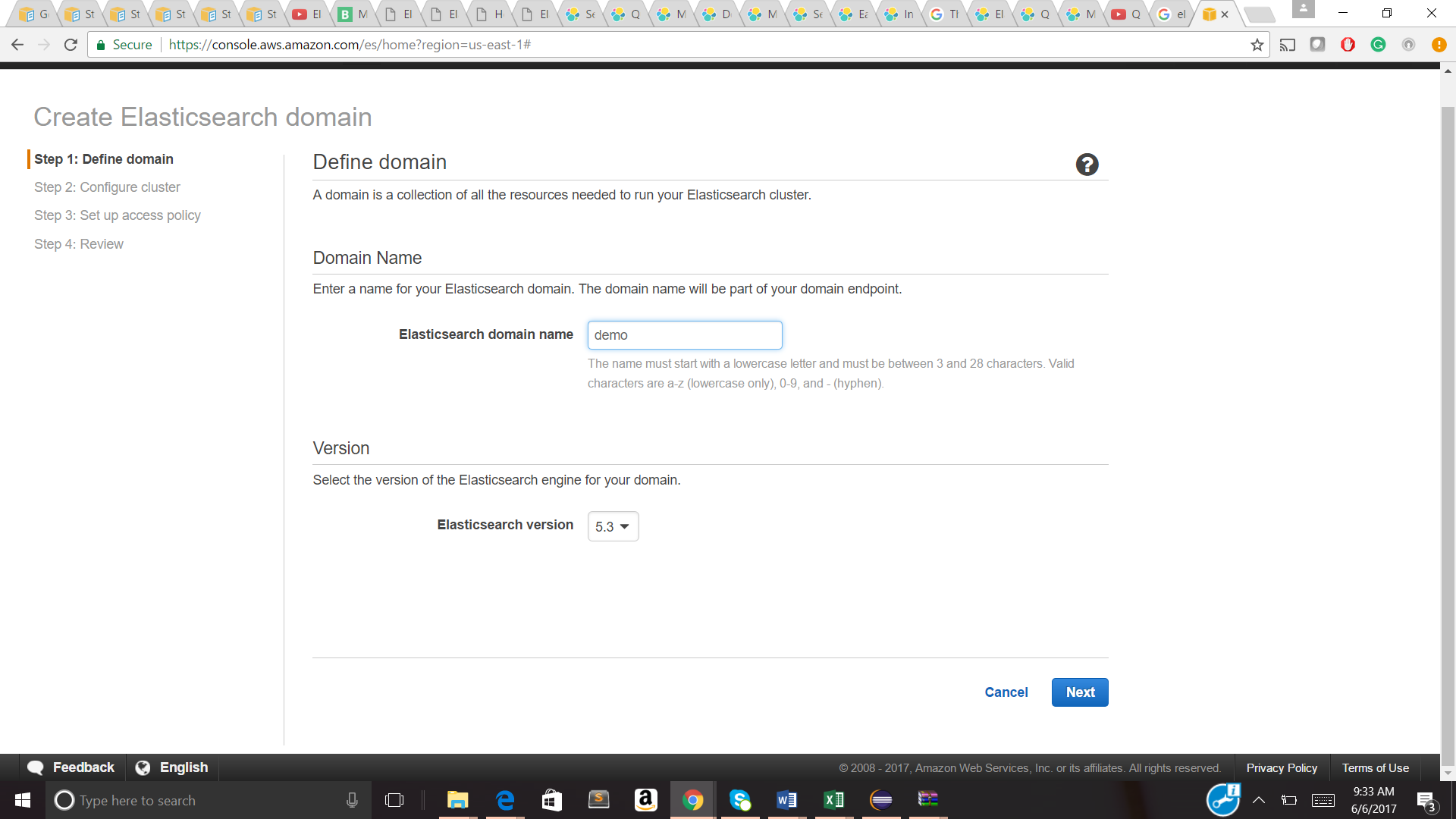
Step - 5: Connected AWS ES with the Lambda Microservice

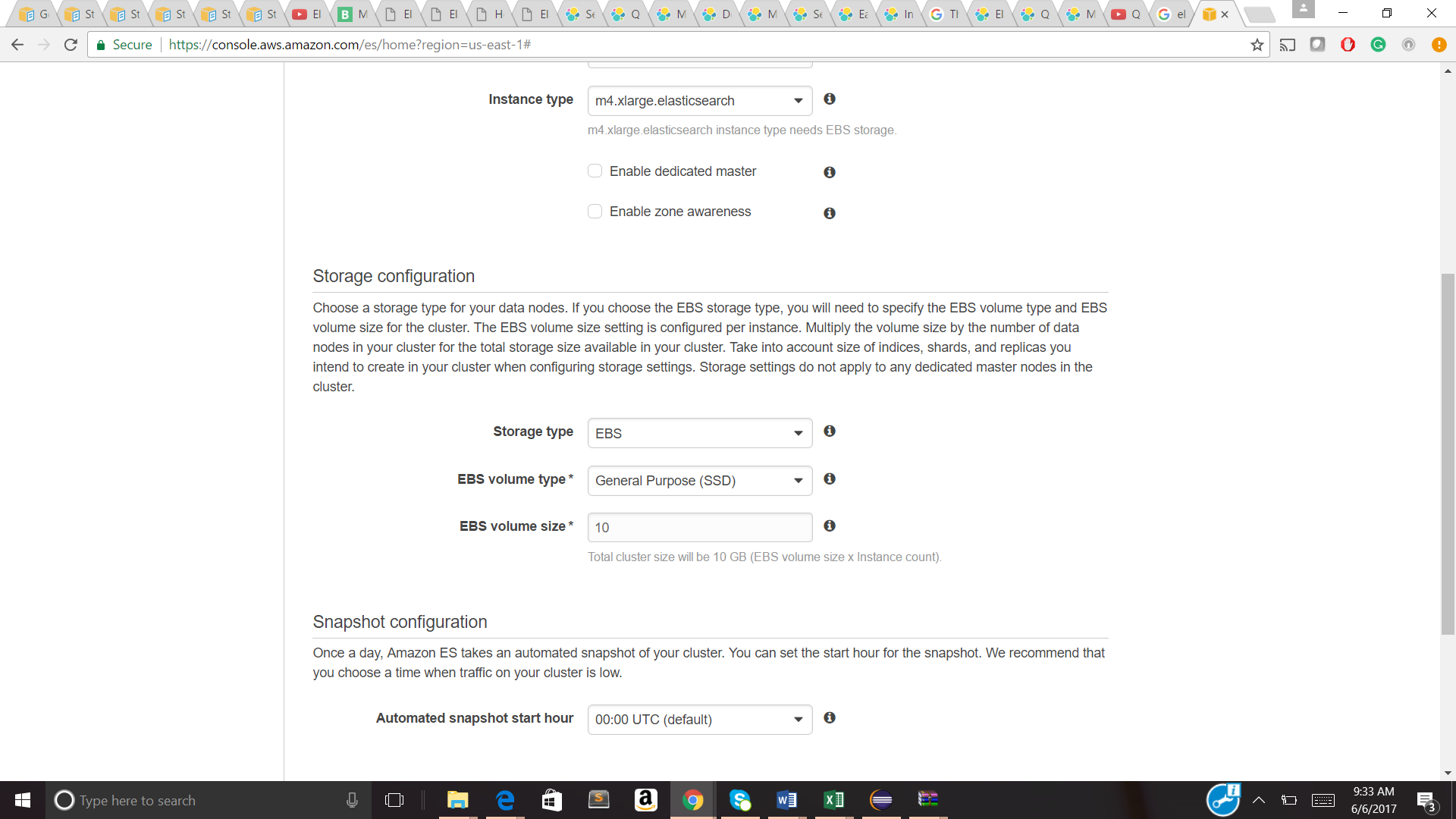
Step - 5: Wrote Lambda handler which calls the functions associated with the search and CRUD operations on the meta data.

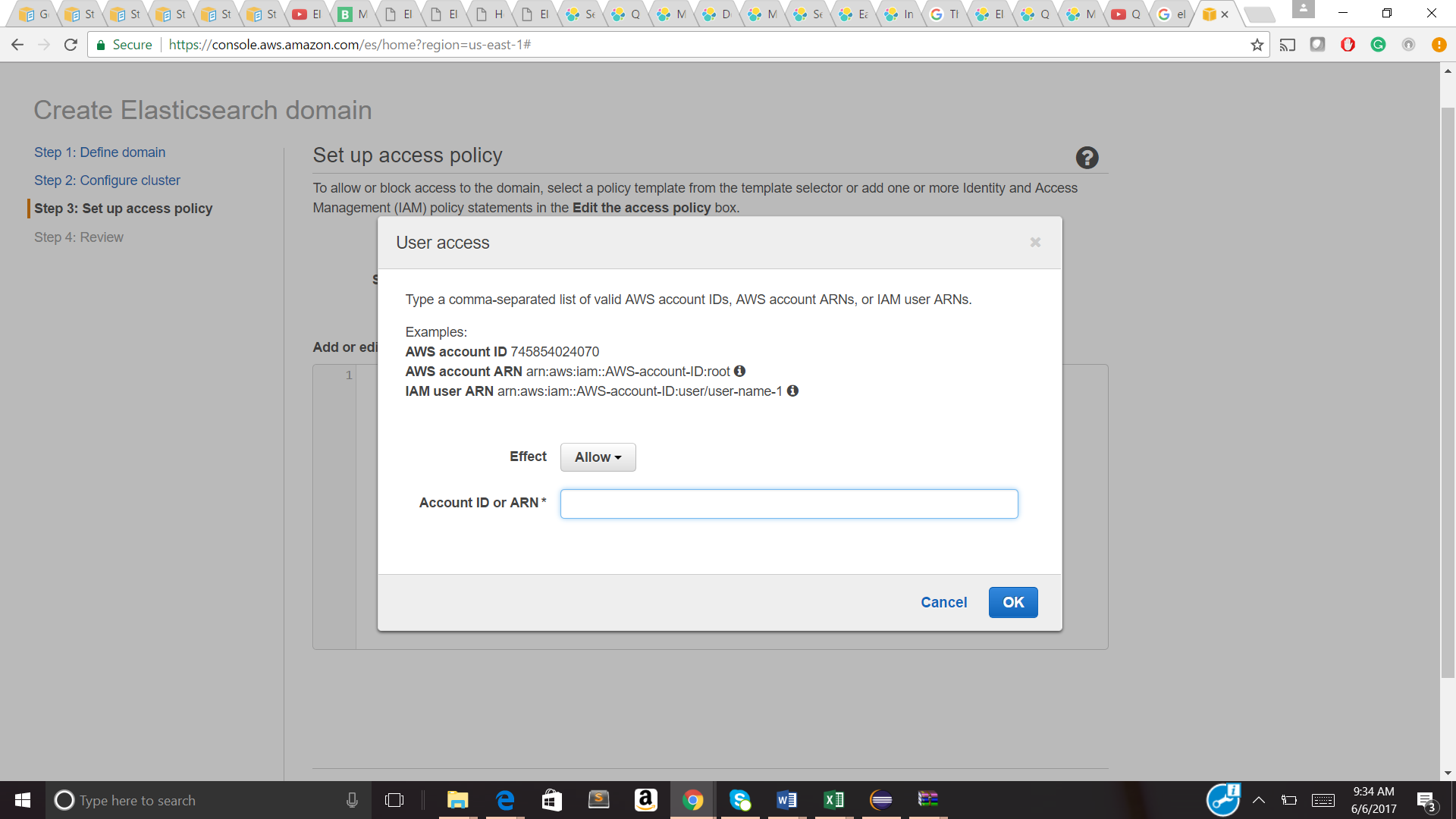
Step - 6: Cluster's health is monitored using AWS Kibana Service

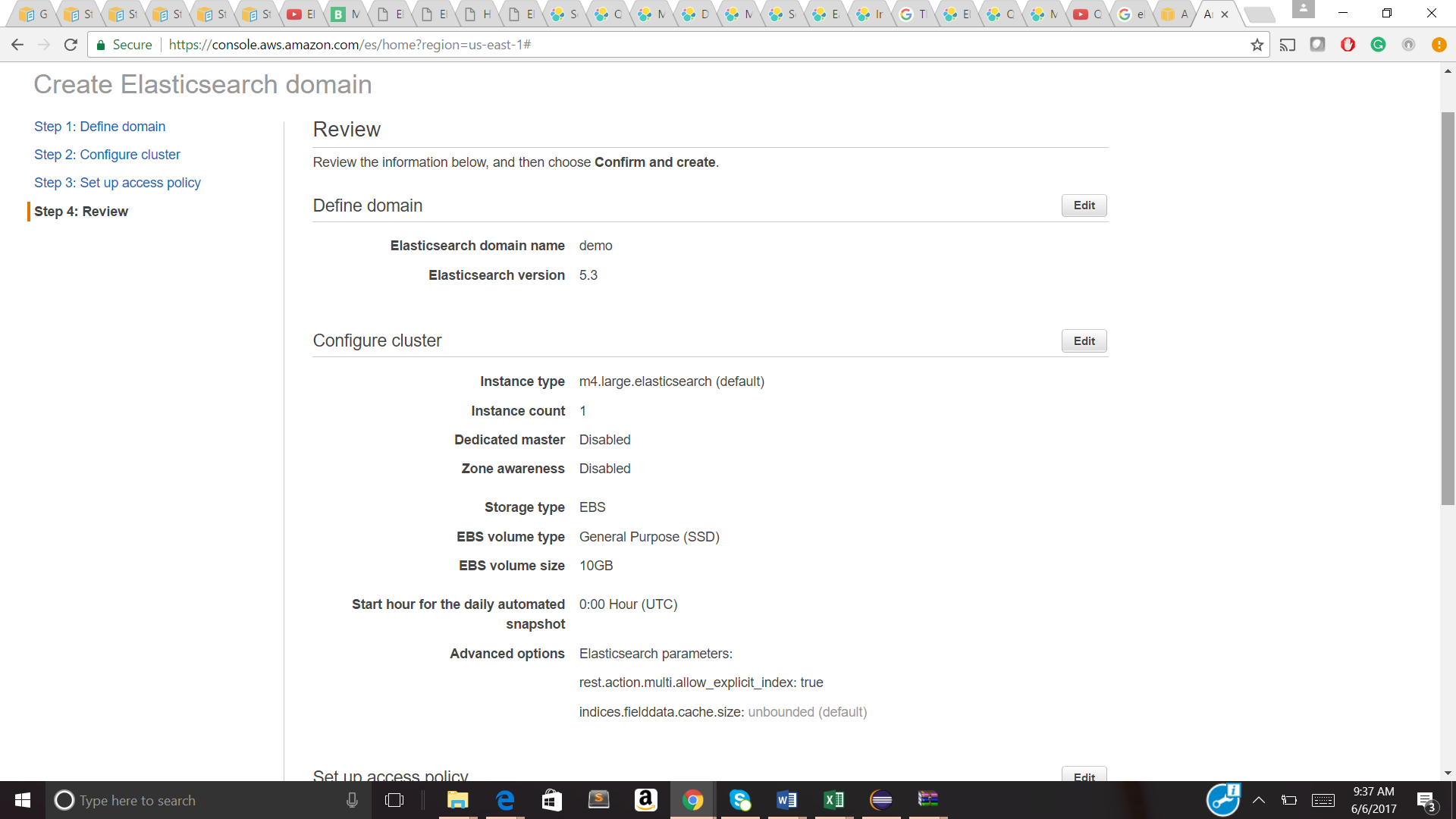
Supporting Screenshots:

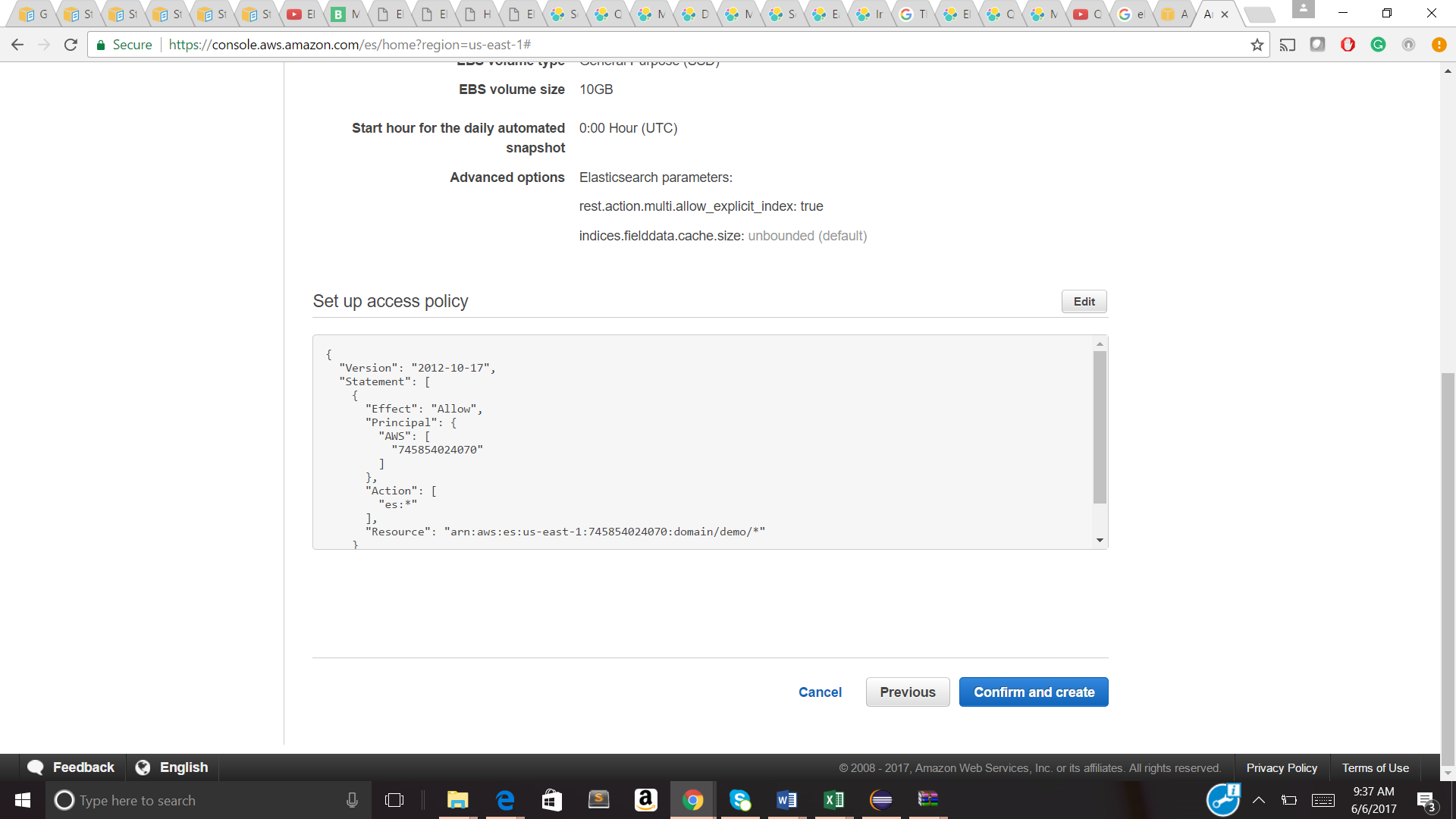


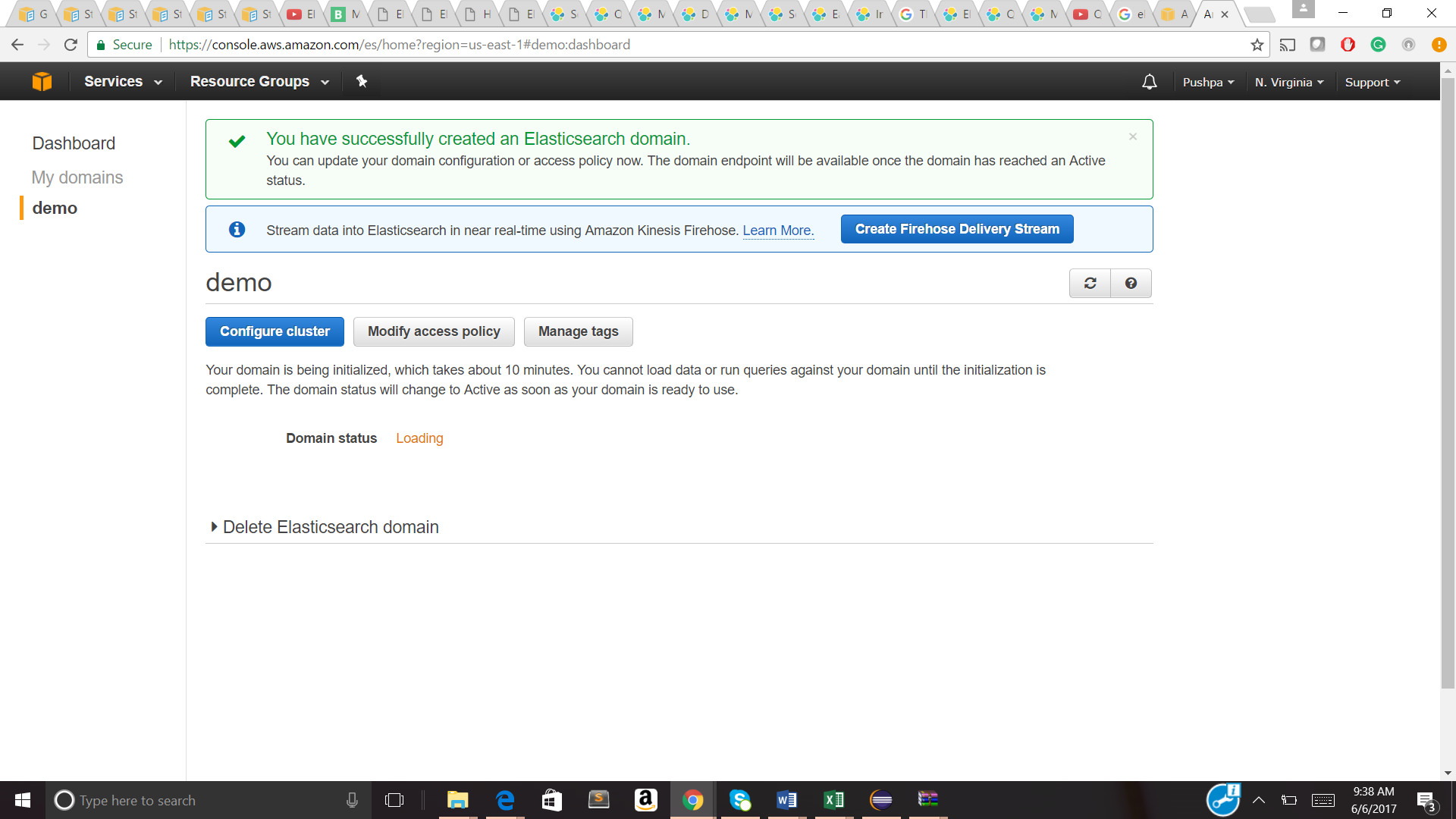


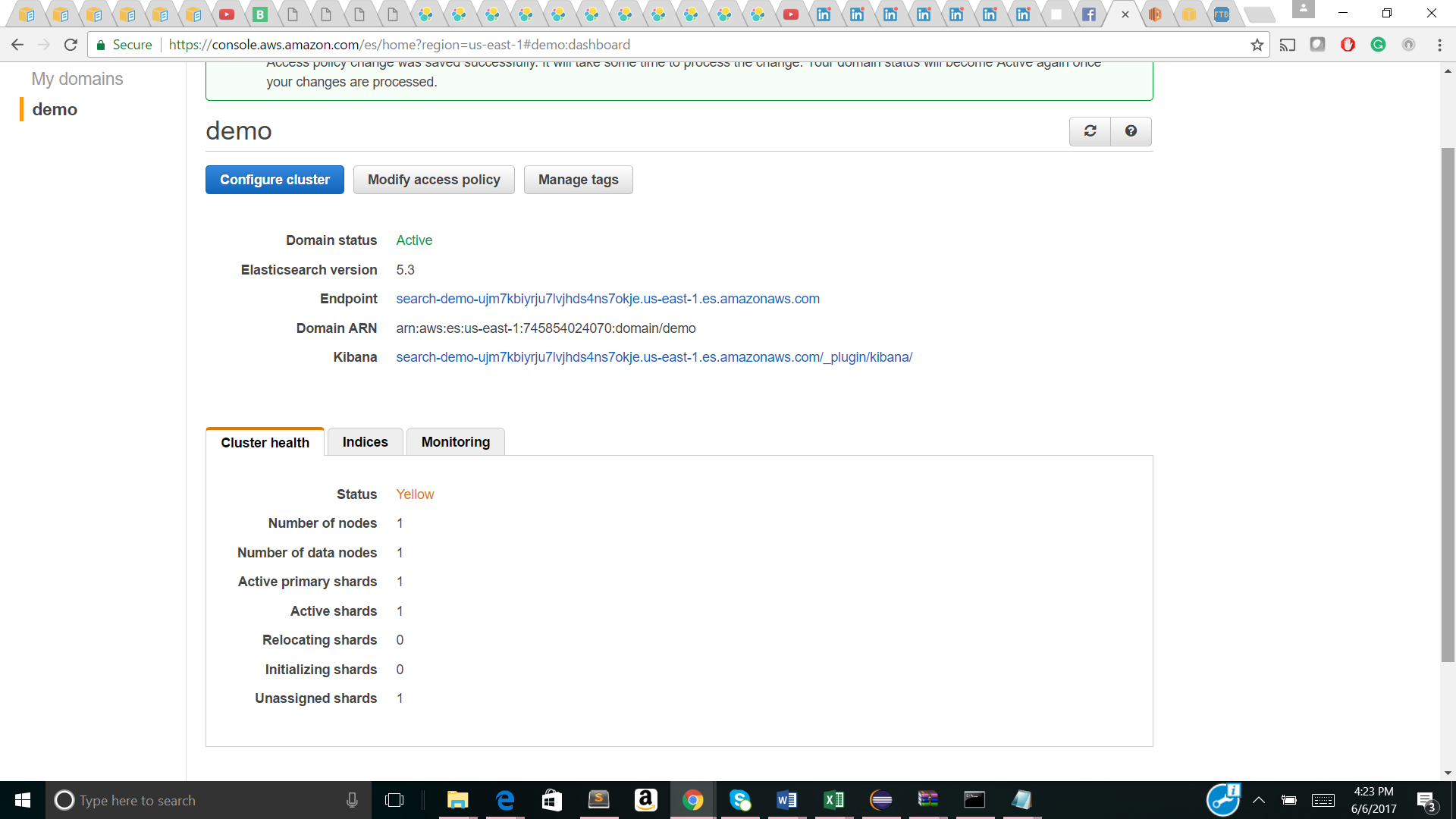


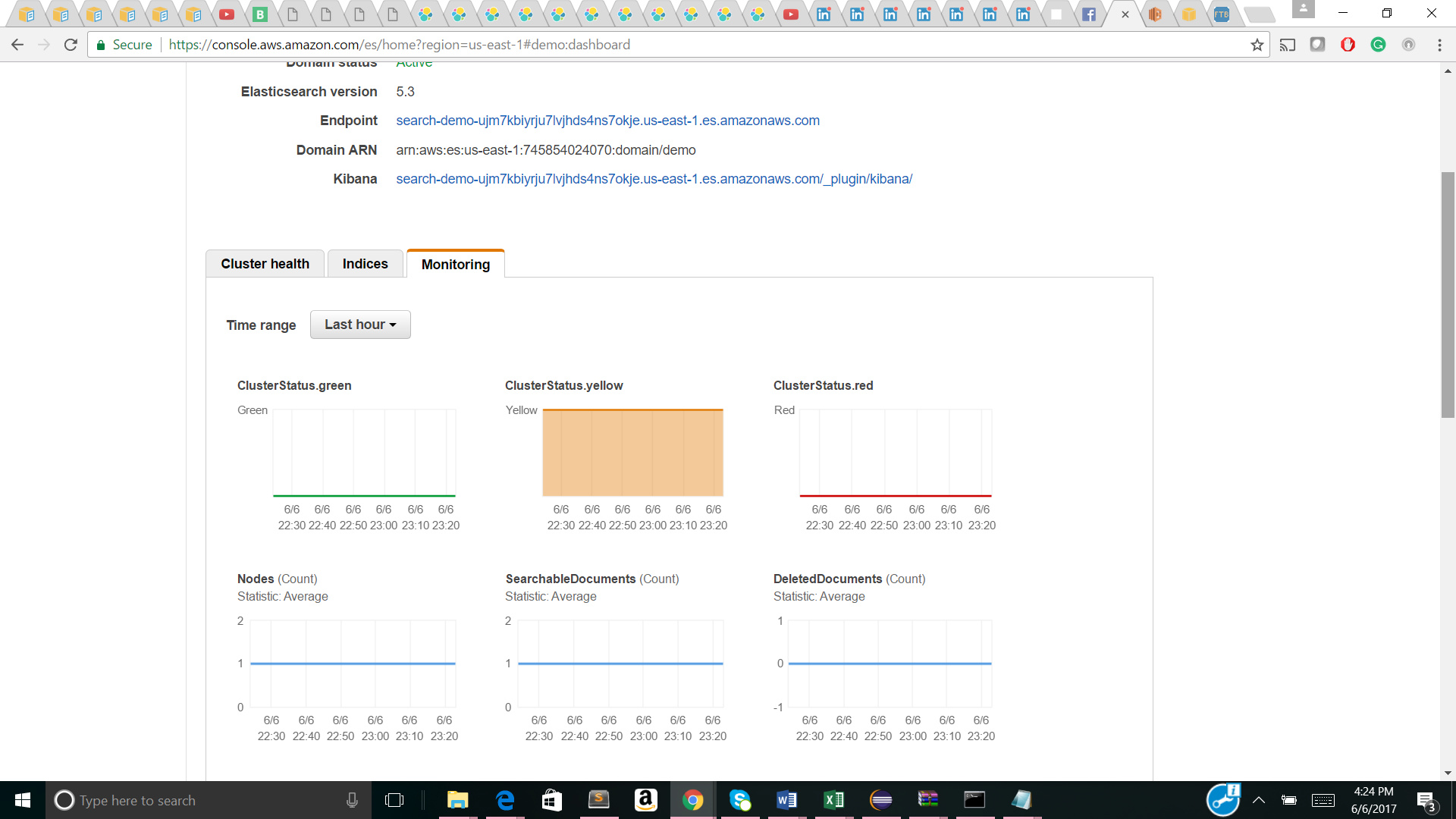


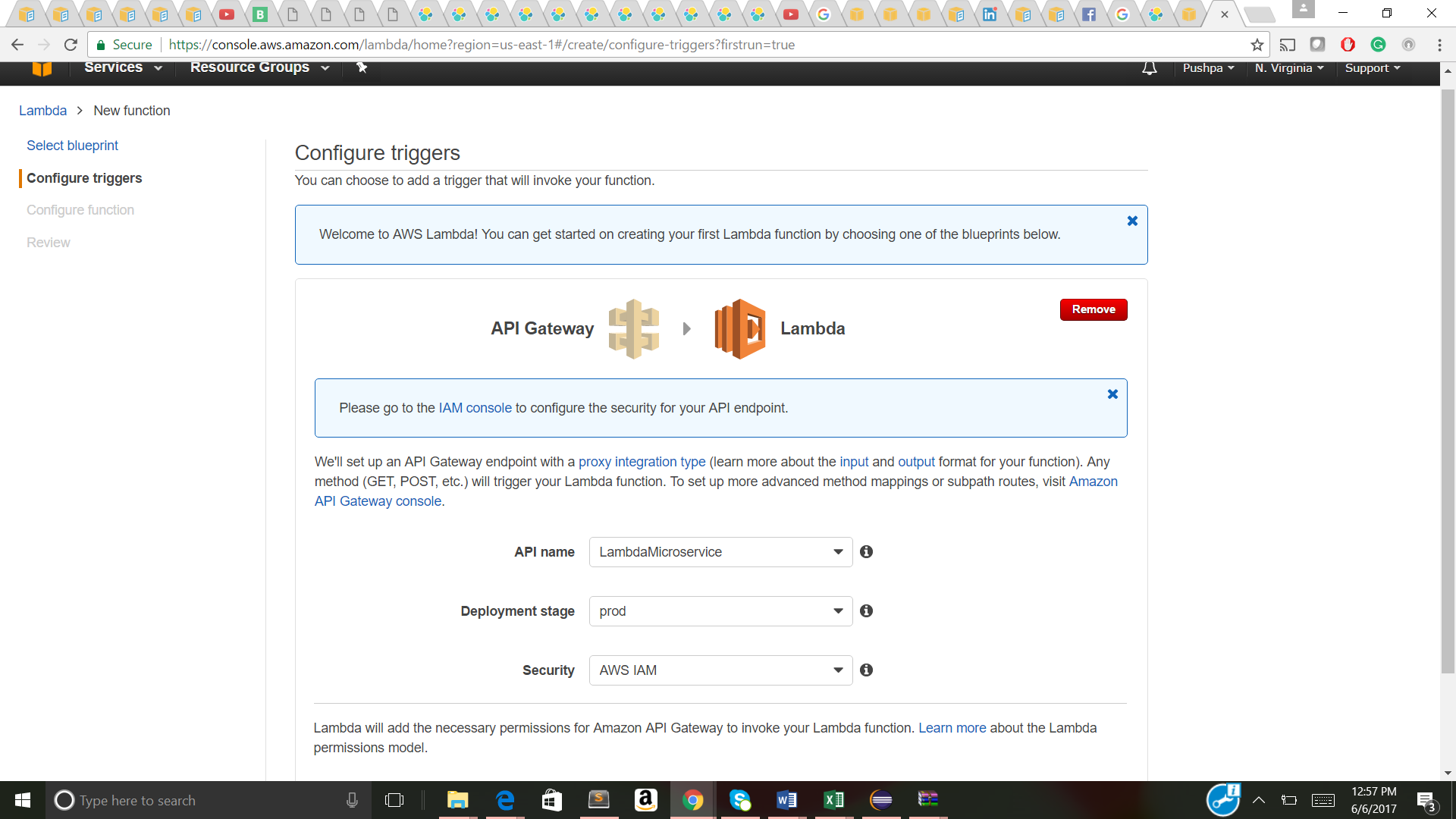


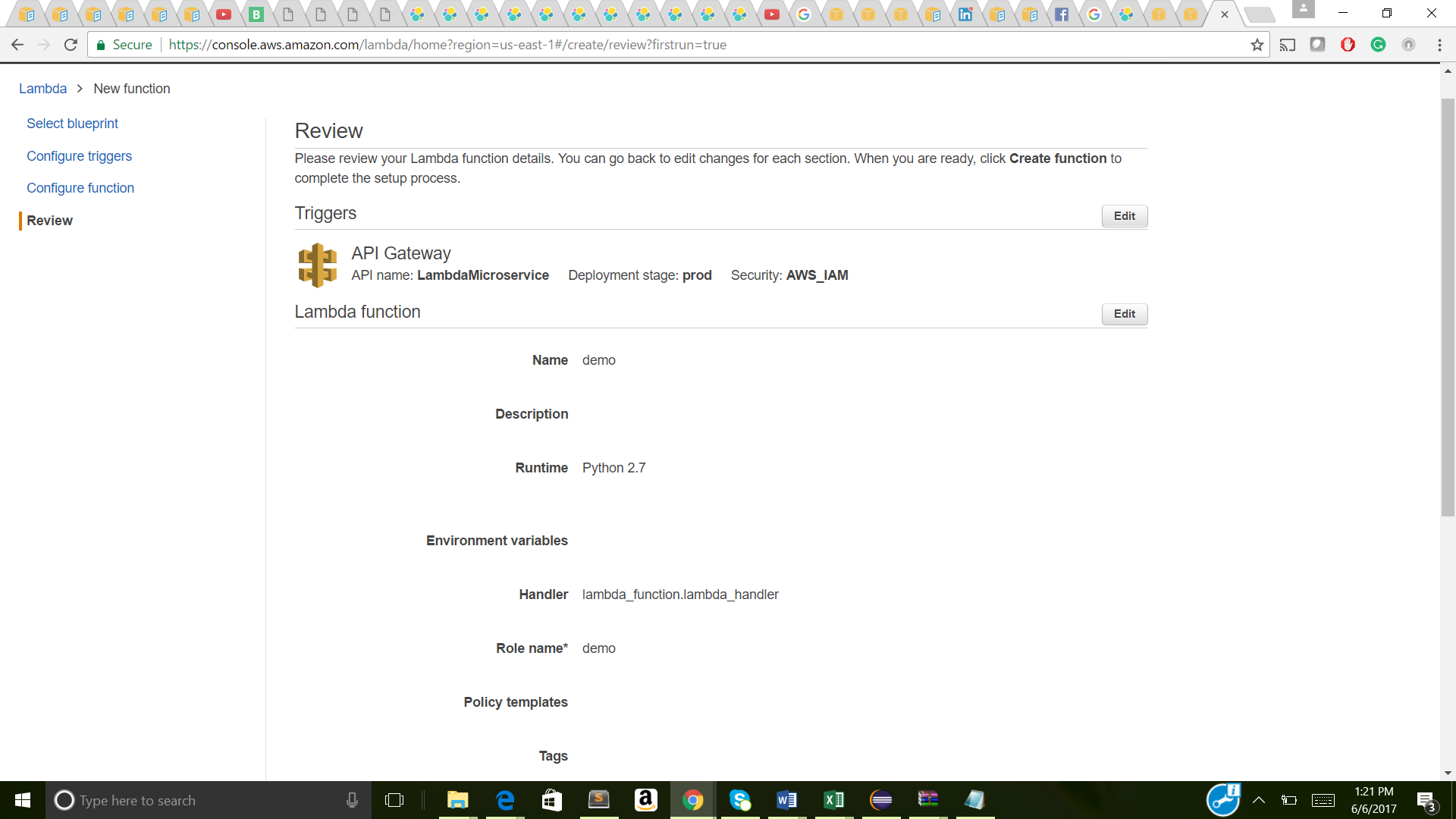


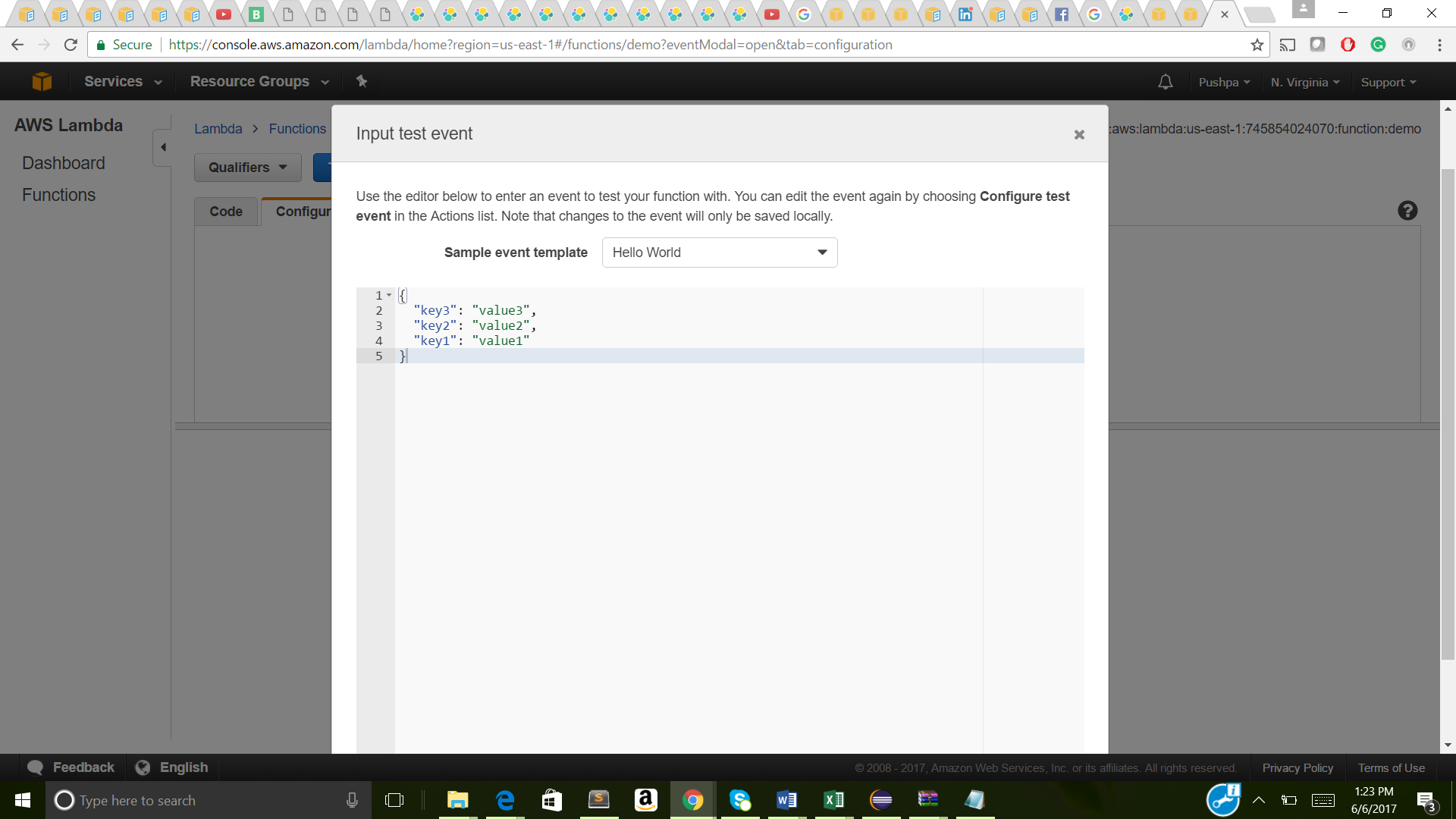


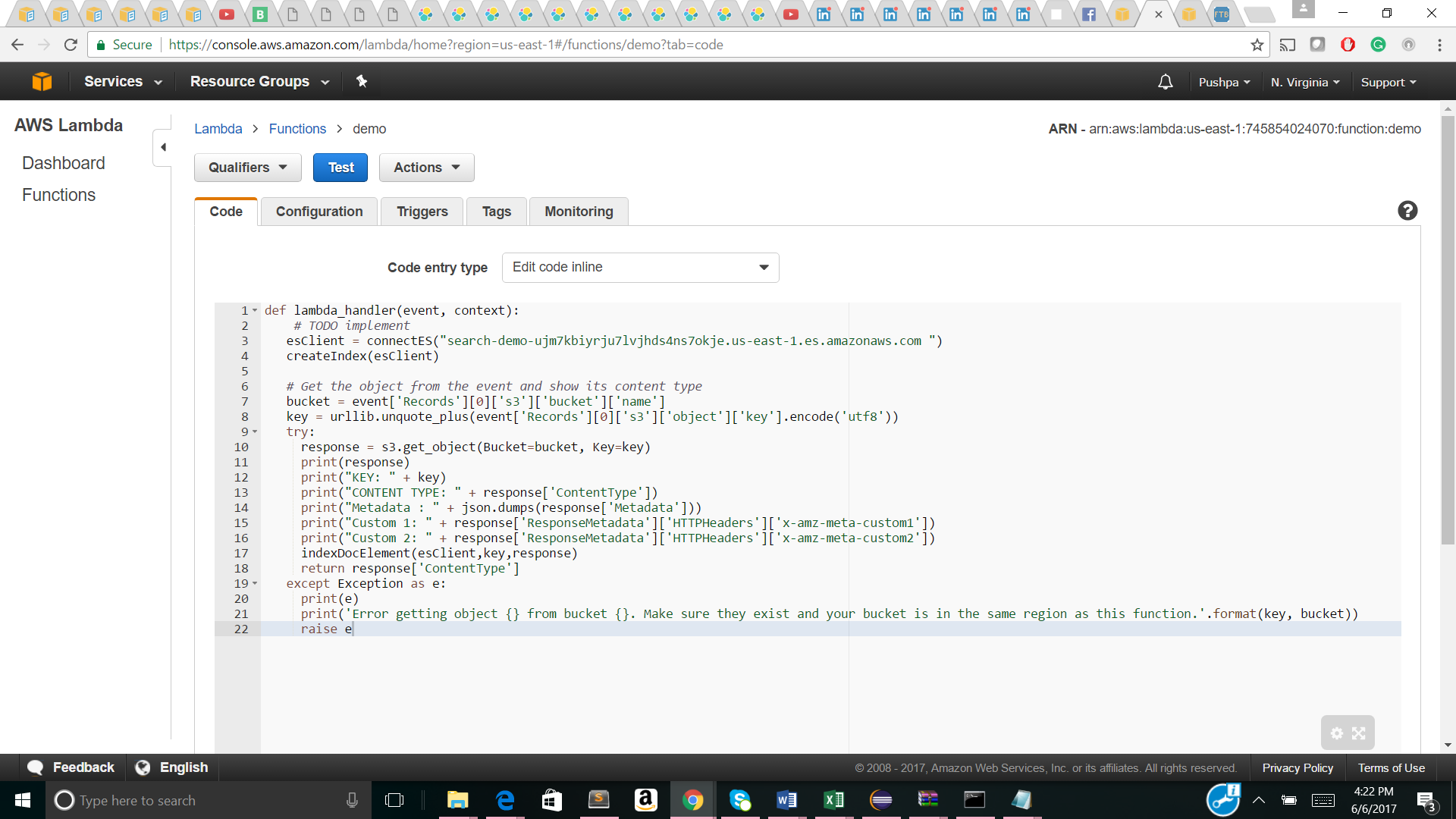


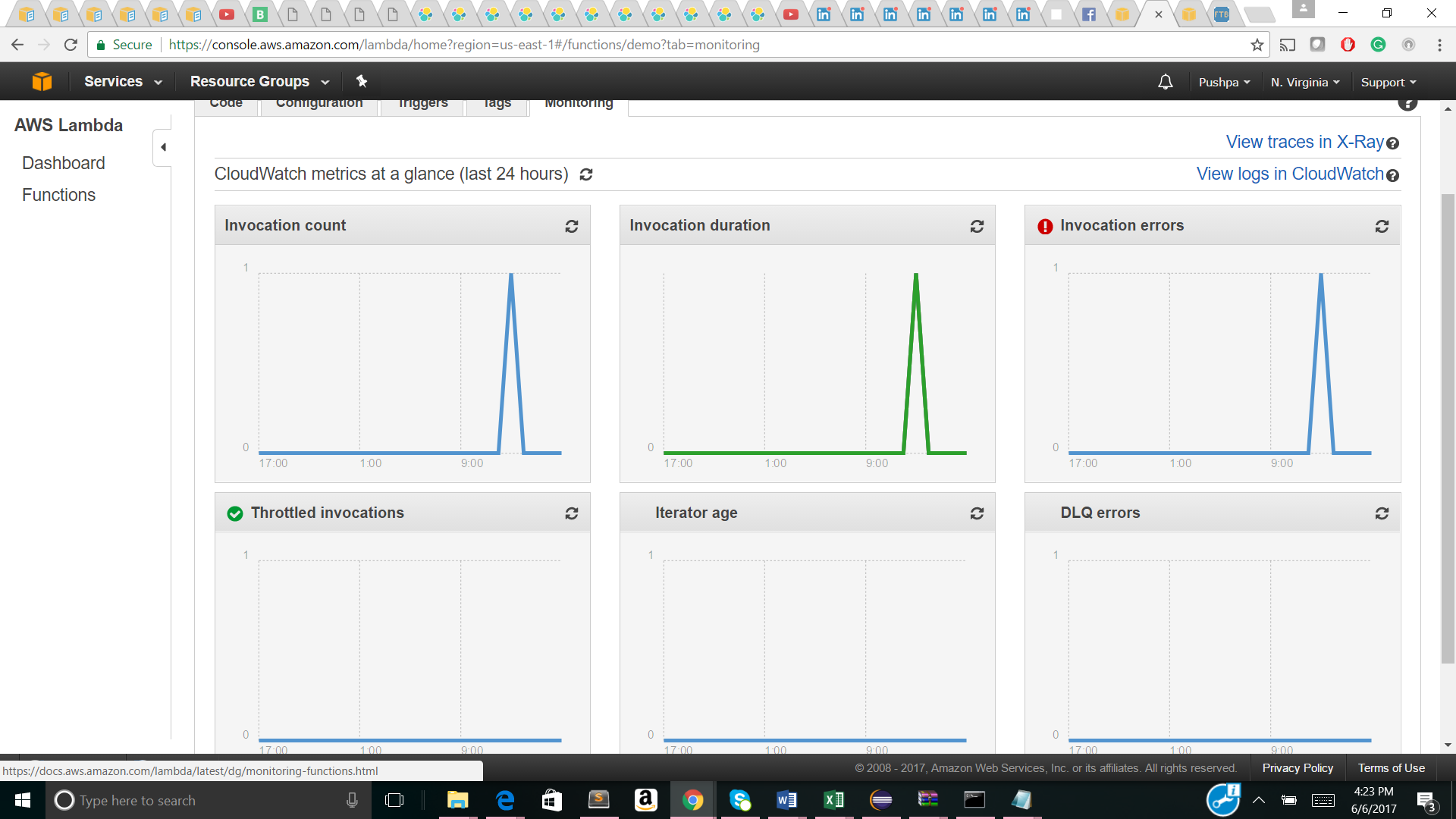












Note: I didn’t make the Account ID, EndPoint and IAM role and ID visible due to security purpose.

References:

* <http://docs.aws.amazon.com/lambda/latest/dg/with-on-demand-https-example-configure-event-source_2.html>
* <https://www.elastic.co/blog/running-elasticsearch-on-aws>
* <https://discuss.elastic.co/t/aws-api-gateway-as-a-proxy-for-elasticsearch/60965/4>
* <http://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-create-resource-and-methods.html>
* <http://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/es-gsg.html>
* http://www.elastichq.org/features.html