DevOps: Week 5

Matthijs Bos
10073558

March 2020

GitHub	`https://github.com/matthijsbos/DevOpsLabWeek5`
Playbook	kubernetes.playbook.yml

1 Experiment

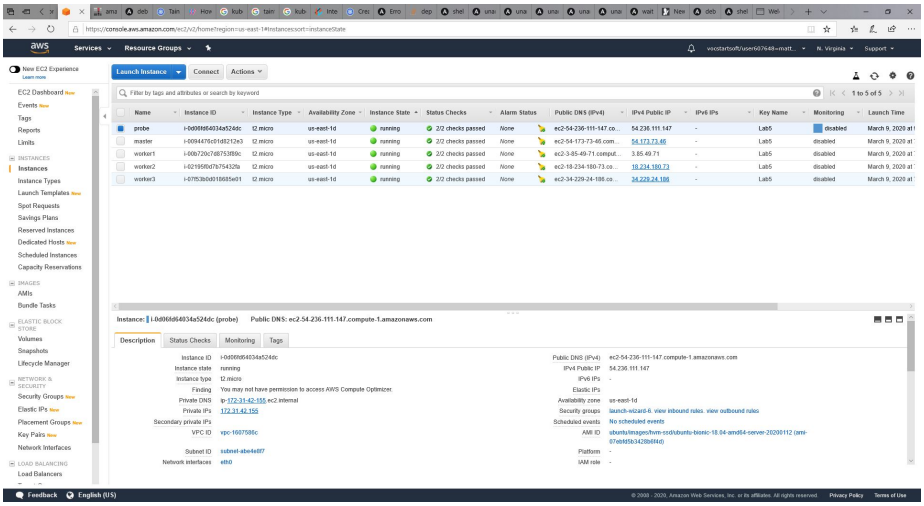

Figure 1: Created a master control node, three worker nodes and an additional node for load testing. The security group inbound traffic rules were relaxed to allow traffic on all ports.

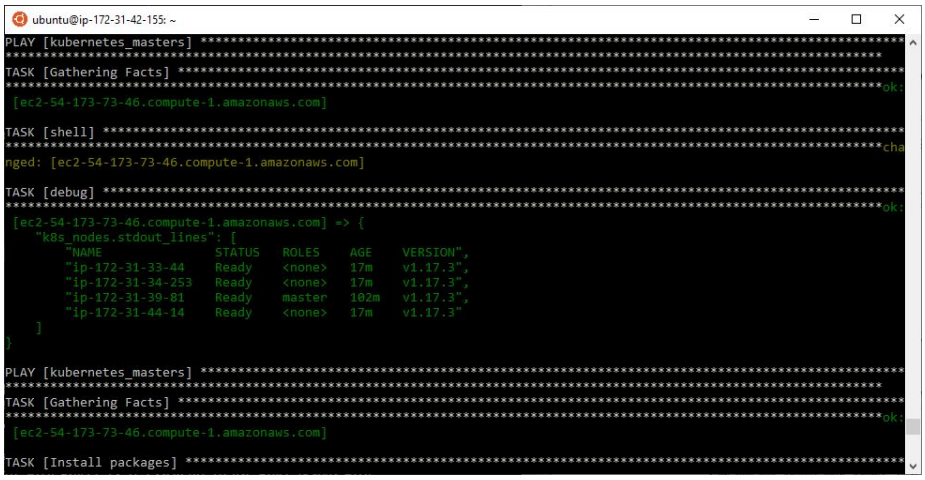

Figure 2: Created playbook to deploy kubernetes cluster

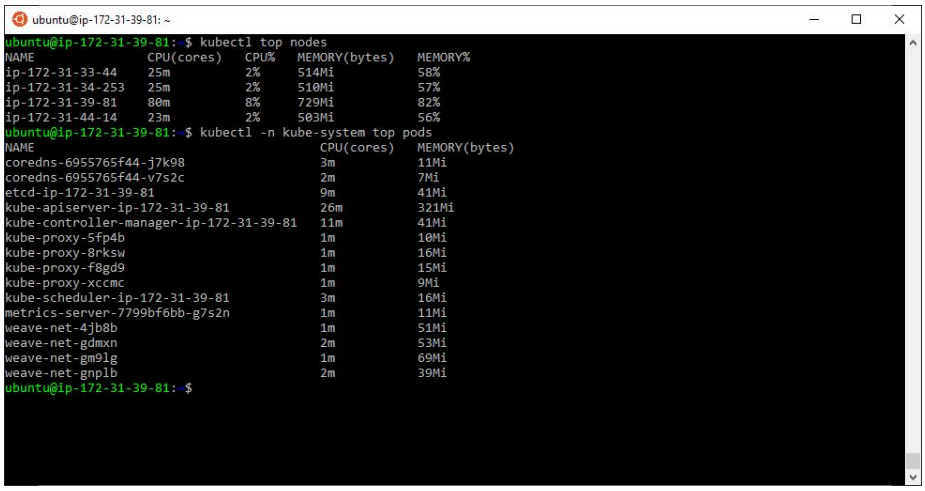

Figure 3: Kubernetes cluster status

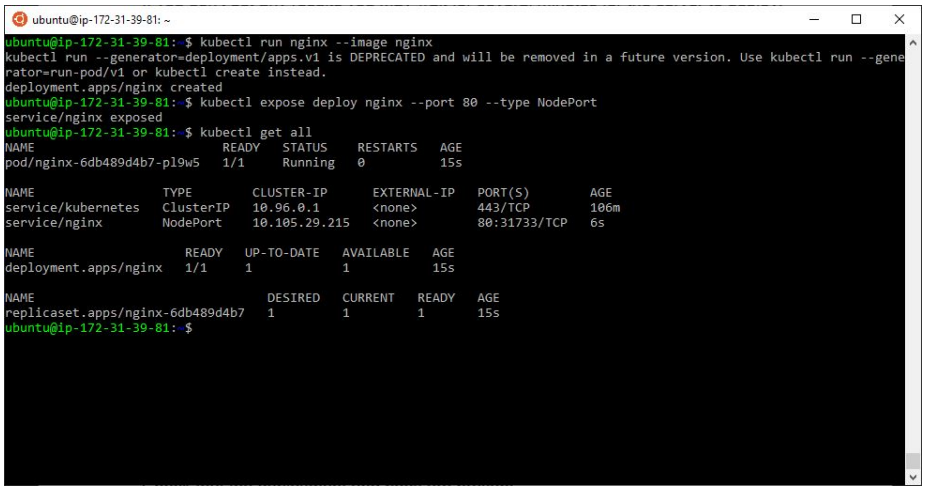

Figure 4: Deploy nginx on cluster

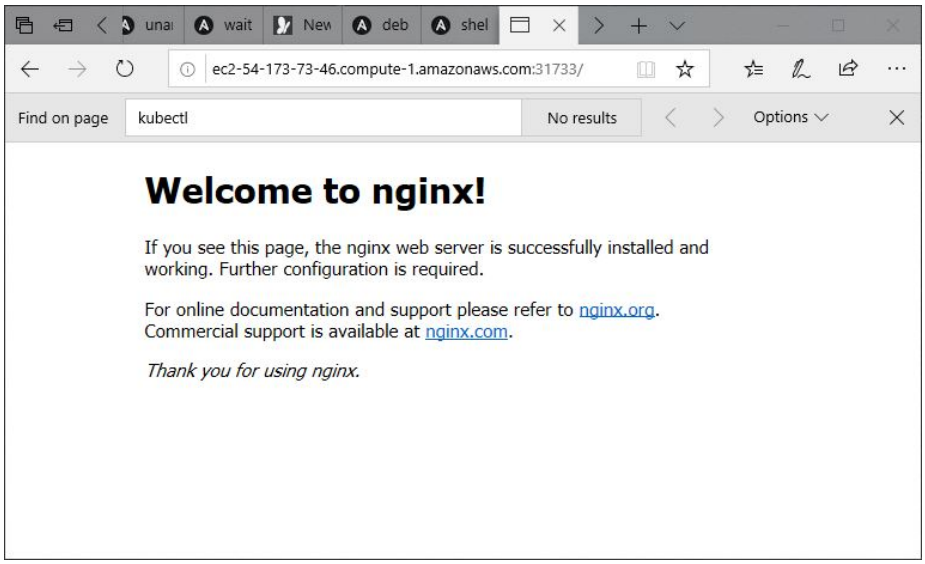

Figure 5: Validated nginx working

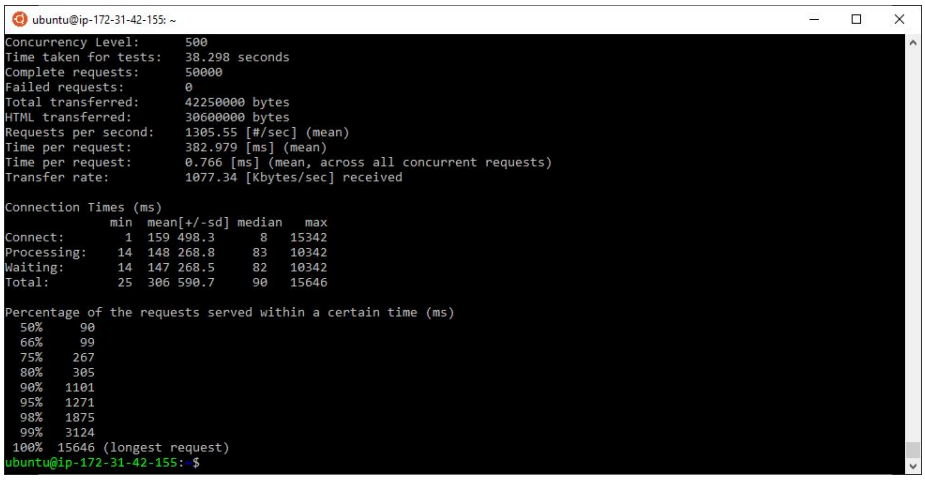

Figure 6: Initial load testing

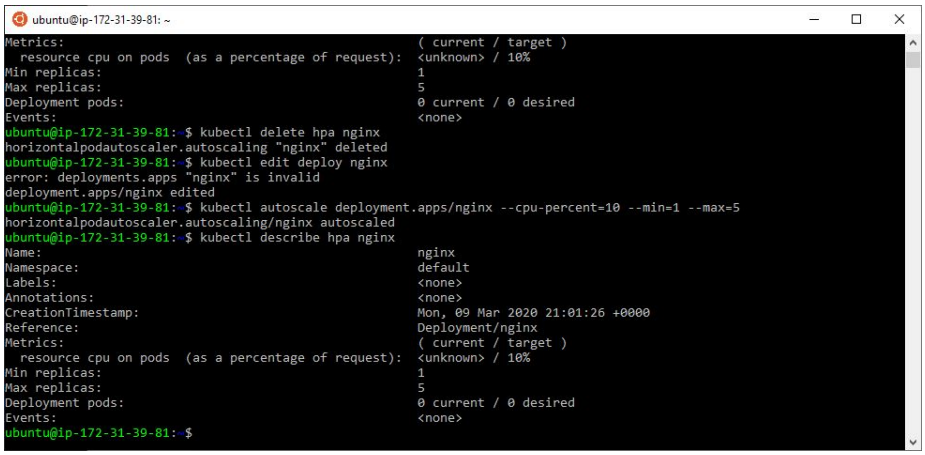

Figure 7: Configured autoscaling

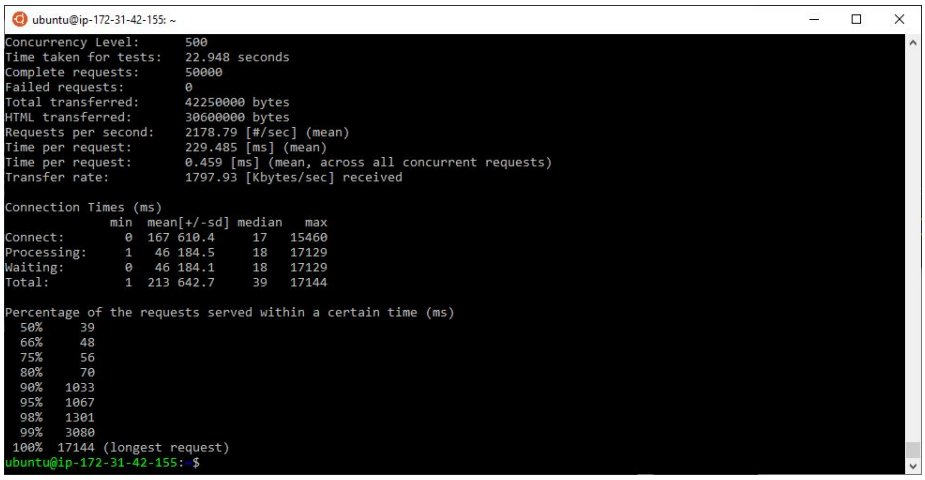

Figure 8: Second load test

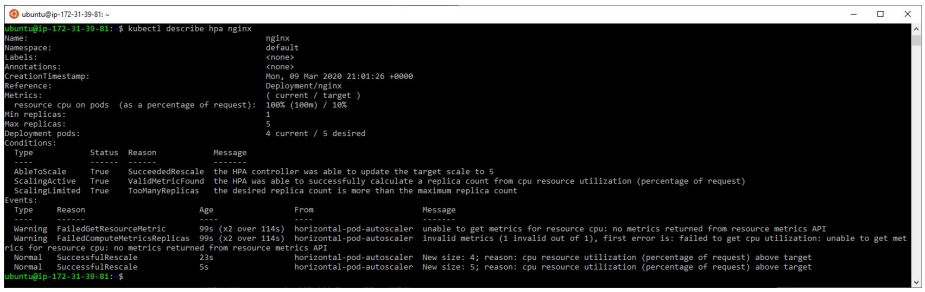

Figure 9: Observed auto scaling events

1.1 Load Testing Observations

In table and figure below, one can clearly observe a significant effect of the auto scaling on the response time during load testing. For up to 80% of the requests, the auto scaling was able to almost minimize the measured effects.

	50%	66%	75%	80%	90%	95%	98%	99%	100%
No auto scaling	90	99	267	305	1101	1271	1875	3124	15646
With auto scaling	39	48	56	70	1033	1067	1301	3080	17144

Table 1: Results for percentage of the requests served within a certain time (ms) with and without auto scaling

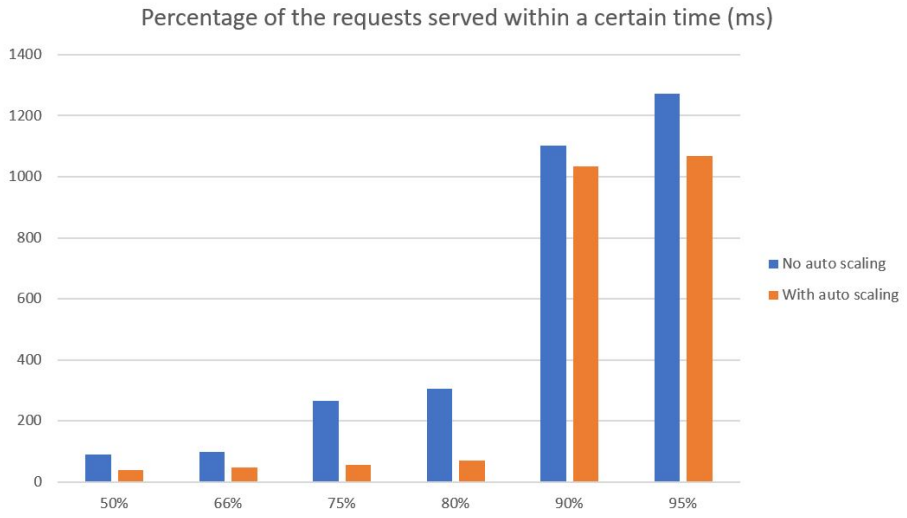

Figure 10: Results for percentage of the requests served within a certain time (ms) with and without auto scaling

2 Self Study Questions

2.1 Ansible & DevOps

Discuss how Ansible can be used during DevOps lifecycle, e.g., which stages? What are the advantages, alternatives of Ansible?

- During the coding phase, an Ansible playbook can be considered a deliverable for implementation of a specific architecture. Playbooks can be checked into revision control for versioning and collaboration.
- During the build phase, Ansible could be used to provision a build toolchain. There may be other tools better suited for this purpose, but the provioning of a Docker development container for example, could very well be automated using Ansible.
- During the tesitng phase, Ansible can be used to provision a testing environment that is a realistic copy of a production environment. Given the possibilities for parameterization of playbook, this allows for easy modification of an existing configuration.
- During the deployment phase, one can use ansible to manage the provisioning of a production environment. This is the primary use case for Ansible.
- During the operation phase, the idempotent characteriscs of Ansible make it very suitable to make modifications on a live production environment.

2.2 Auto Scaling

Discuss the benefits of using auto scaling for Cloud applications, and for DevOps? Based on the experiments, discuss other scenarios where autoscaling can be used?

Auto scaling is especially useful during operation of a production environment where the workload of managing the number of workers can be automated. The quality of service can automatically be managed based on preset parameters.

Auto scaling can be usefull for any kind of workload that allows for a horizontal growth by dynamically growing and shrinking the pool of workers. This can be the case for front-end servers such as in the experiement, but can also be done for background workers, for example. One example can be a auto scaling pool of background workers that scale based on the number of pending work items in a queue.

2.3 Ansible & Azure

Can you use Ansible with Azure? Is it essential to use Azure DevOps with Ansible?

Ansible is perfectly suitable for use with Azure, since it doesn't put any requirements on the target hosts other than that these should be accessible through a remote terminal, preferably SSH. There are also specific Ansible modules that interact with Azure APIs for configuration of resources other than compute hosts.

Ansible DevOps doesn't require or restrict the use of Ansible. Azure DevOps is built on top of the regular Azure services that can be provisioned using Ansible.

2.4 Azure Resource Manager

Do you have to use Azure Resource Manager (ARM) when working with Azure resources and deploying VM and services on Azure cloud? Briefly compare ARM and CloudFormation.

ARM and Cloudformation serve identical purposes in the sense that they are platform-specific methods for provising and configuration of cloud resources. Azure and AWS both offer APIs as well, allowing for external tools to be used as an alternative, not limiting end users to use either of both.

ARM and Cloudformation primarily differ in syntax and used library constructs, but the underlying concepts are the same. Someone switching between the two will mainly be investing in acquiring knowledge on the specific services and configuration options for the different cloud platforms.