Forum FY20 Nginx Controller Documentation

Matthieu DIERICK

Apr 14, 2020

Contents:

CHAPTER 1

Publish and protect modern applications

Warning: For any remark or mistake in this lab, please send a Teams chat to Matthieu DIERICK.

Note: Contributors : Philippe Cloup, Nicolas Ménant, Fouad Chmainy

1.1 Intro - Understand the infrastructure and the workflow

Welcome into the Forum FY20 breakout demo "Publish and Protect Modern Applications".

Warning: For any remark or mistake in this lab, please send a Teams chat to Matthieu DIERICK.

First of all, have a look on the PowerPoint deck.

This section convers the concept and the goals of the breakout.

Audiance:

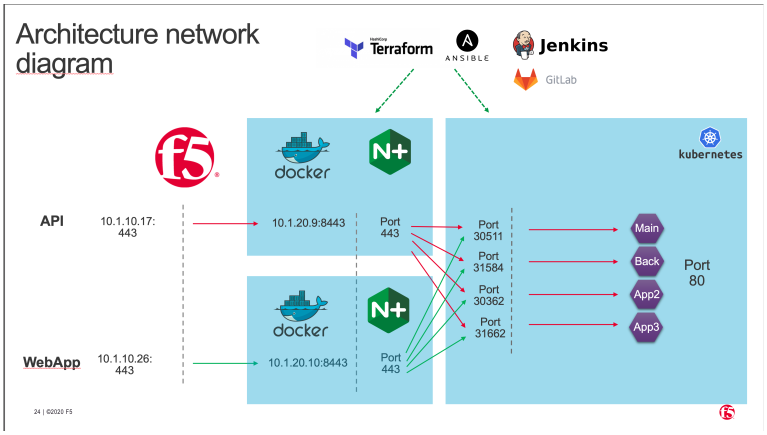
- DevOps
- Netops

Goals:

- Demonstrate Nginx and BIG-IP are part of application lifecyle
- Demonstrate Nginx and BIG-IP are fully integrated with CI/CD pipeline
- Demonstrate the new Controller 3.x capabilities and automation tool set
- Demonstrate Declarative WAF

1.1.1 Architecture of the demo

The diagram below explains the application routes.



Architecture network diagram

1.1.2 Workflow of the demo

The demo is splitted in 4 steps :

- Deploy and publish Arcadia Main App
- Deploy and publish Money Transfer App
- Deploy and publish Refer Friends App
- Apply WAF policy

Step 1 - DevOps deploy Arcadia application

Note: Goal is to use the GUI in the NGINX Controller for our traditional customers. NetOps will configure the services (MainApp and BackEnd) manually.

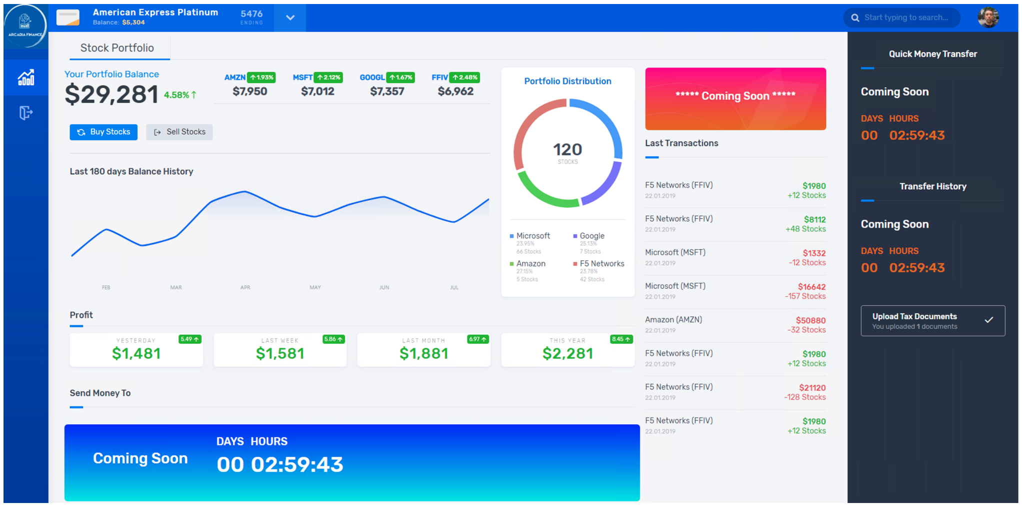
Tasks:

1. DevOps commit a new code in GitLab in order to publish a brand new application "Arcadia Bank"
2. **GitLab webhooks this commit and ask Jenkins to run a pipeline. This pipeline:**

   1. Deploy Arcadia application in Kubernetes (Terraform).
   2. Deploy nodeports in Kubernetes (but it could be KIC) (Terraform).
   3. Deploy NGINX+ instances (ADC) in Docker, in front of this K8S cluster (Terraform)
   4. Create Gateways in NGINX Controller for each NGINX+ instance (Ansible)
   5. Deploy AS3 template into front BIGIP to publish publically the application - without WAF (Ansible)

3. **NetOps create ADC configuration in NGINX controller in order to "route" traffic to the right K8S service**

   1. MainApp (/*) to service MainApp
   2. BackEnd (/file*) to service BackEnd



F5 Networks (FFIV) 22.01.2019	$1980 +12 Stocks
F5 Networks (FFIV) 22.01.2019	$8112 +48 Stocks
Microsoft (MSFT) 22.01.2019	$1332 -12 Stocks
Microsoft (MSFT) 22.01.2019	$16642 -157 Stocks
Amazon (AMZN) 22.01.2019	$50880 -32 Stocks
F5 Networks (FFIV) 22.01.2019	$1980 +12 Stocks
F5 Networks (FFIV) 22.01.2019	$21120 -128 Stocks
F5 Networks (FFIV) 22.01.2019	$1980 +12 Stocks

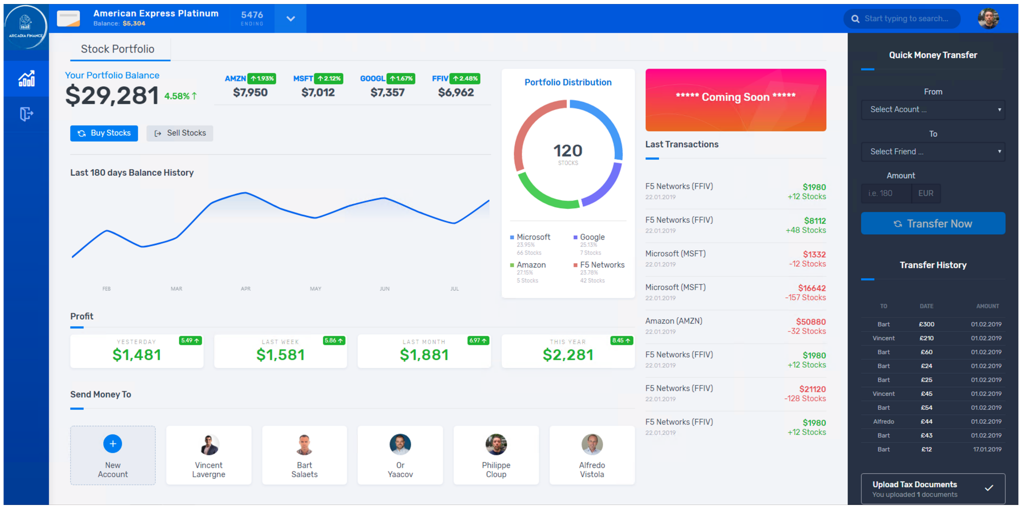
> **Warning:** At this stage, the first part of the application is published and can be accessed and demonstrated. We can see Money Transfert application is not yet there, same for Refer Friends.

Step 2 - DevOps deploy Money Transfer application

Note: Goal is to demonstrate NGINX Controller has a REST API to configure objects. NetOps will configure the service (Money Transfer) via REST API.

Tasks:

1. DevOps commit a new code in GitLab in order to publish the second part of the Arcadia Bank website. This new application allows money transfer between friends.
2. **GitLab webhooks this commit and ask Jenkins to run a pipeline. This pipeline:**
   1. Deploy Money Transfer application in Kubernetes (Terraform)
   2. Deploy nodeports in Kubernetes (Terraform)
3. NetOps use REST API to publish this new app on NGINX+ instances



Last Transactions

F5 Networks (FFIV) 22.01.2019	$1980 +12 Stocks
F5 Networks (FFIV) 22.01.2019	$8112 +48 Stocks
Microsoft (MSFT) 22.01.2019	$1332 -12 Stocks
Microsoft (MSFT) 22.01.2019	$16642 -157 Stocks
Amazon (AMZN) 22.01.2019	$50880 -32 Stocks
F5 Networks (FFIV) 22.01.2019	$1980 +12 Stocks
F5 Networks (FFIV) 22.01.2019	$21120 -128 Stocks
F5 Networks (FFIV) 22.01.2019	$1980 +12 Stocks

Transfer History

TO	DATE	AMOUNT
Bart	£300	01.02.2019
Vincent	£210	01.02.2019
Bart	£60	01.02.2019
Bart	£24	01.02.2019
Bart	£25	01.02.2019
Vincent	£45	01.02.2019
Bart	£54	01.02.2019
Alfredo	£44	01.02.2019
Bart	£43	01.02.2019
Bart	£12	17.01.2019

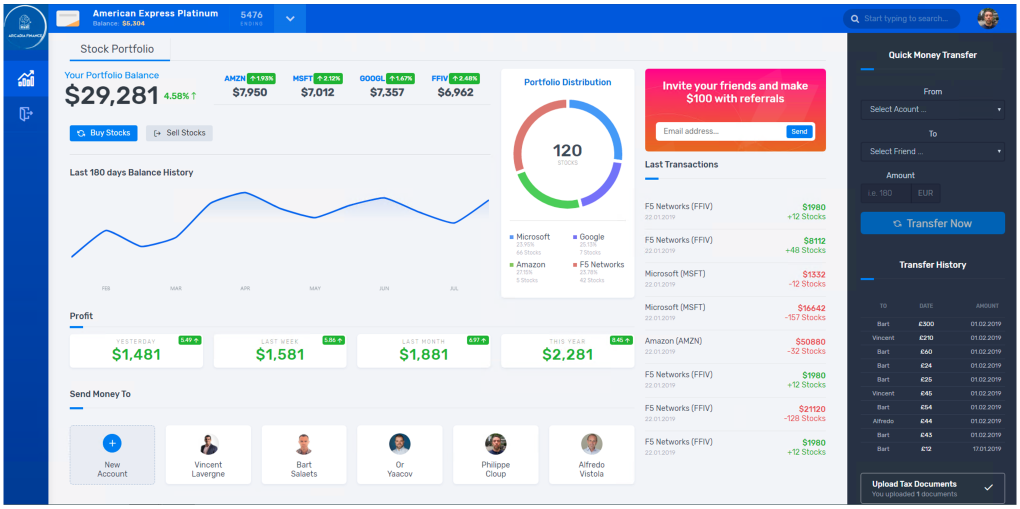
Warning: At this stage, the Money Transfer application is published and can be accessed and demonstrated

Step 3 - DevOps deploy Refer Friends application

Note: Goal is to demonstrate NGINX Controller can be part of the application lifecycle and CICD. NetOps don't configure anything.

Tasks:

1. DevOps commit a new code in GitLab in order to publish the third and last part of the Arcadia Bank website. This new application allow a customer to refer friends with their email address.
2. **GitLab webhooks this commit and ask Jenkins to run a Pipeline. This pipeline:**
    1. Deploy Refer Friends application in Kubernetes (Terraform)
    2. Deploy nodeports in Kubernetes (Terraform)
    3. Configure all components in NGINX Controller (Ansible)



Last Transactions

F5 Networks (FFIV) 22.01.2019	$1980 +12 Stocks
F5 Networks (FFIV) 22.01.2019	$8112 +48 Stocks
Microsoft (MSFT) 22.01.2019	$1332 -12 Stocks
Microsoft (MSFT) 22.01.2019	$16642 -157 Stocks
Amazon (AMZN) 22.01.2019	$50880 -32 Stocks
F5 Networks (FFIV) 22.01.2019	$1980 +12 Stocks
F5 Networks (FFIV) 22.01.2019	$21120 -128 Stocks
F5 Networks (FFIV) 22.01.2019	$1980 +12 Stocks

Transfer History

TO	DATE	AMOUNT
Bart	£300	01.02.2019
Vincent	£210	01.02.2019
Bart	£60	01.02.2019
Bart	£24	01.02.2019
Bart	£25	01.02.2019
Vincent	£45	01.02.2019
Bart	£54	01.02.2019
Alfredo	£44	01.02.2019
Bart	£43	01.02.2019
Bart	£12	17.01.2019

Warning: At this stage, the Refer Friends application is published and can be accessed and demonstrated. The Arcadia Bank website is finished, but not yet secured.

Step 4 - NetOps/SecOps publish WAF policy to protect Arcadia application

Note: Goal is to demonstrate BIG-IP Advanced WAF has a Declrarative API interface to push WAF policies.

Task:

1. NetOps run a Jenkins pipeline that will push a new AS3 declaration with a WAF policy built by Secops

Warning: At this stage, the Arcadia Bank website is published and secured.

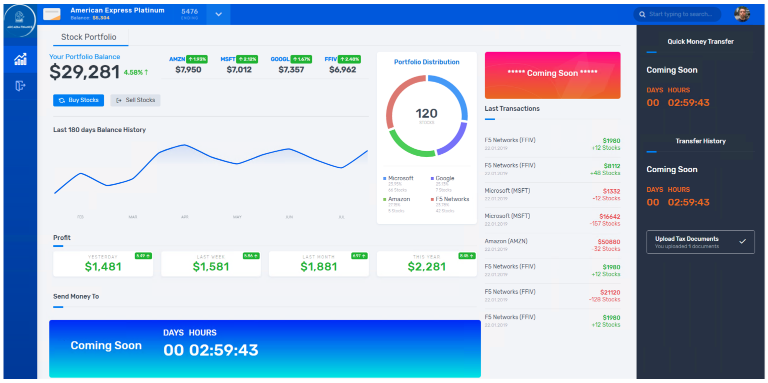
1.2 Demo - Run the demo in UDF

In this section, I demonstrate all the step to deliver the demo

1.2.1 Step 1 - DevOps deploy Arcadia Application

In this module, we will deploy the 2 main containers for Arcadia Bank application and we will publish them.

Note: At the end of this module, Arcadia Bank application will look like this.



Warning: Please keep case senstive for all objects below

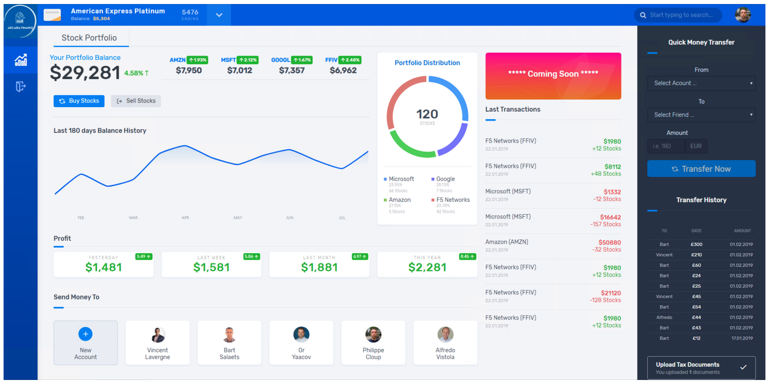
1. Application name : app_webapp
2. Components:
    1. cp_mainapp:
        1. Ingress URI: http://www.arcadia-finance.io/
        2. Workload: wl_mainapp
            1. URI: http://mainapp.nginx-udf.internal:30511
    2. cp_back:
        1. Ingress URI: http://www.arcadia-finance.io/files/
        2. Workload: wl_backend
            1. URI: http://backend.nginx-udf.internal:31584

Video of this module :

1.2.2 Step 2 - DevOps deploy Money Transfer application

In this module, we will deploy the Money Tranfer container for Arcadia Bank application and we will publish it.

Note: At the end of this module, Arcadia Bank application will look like this.

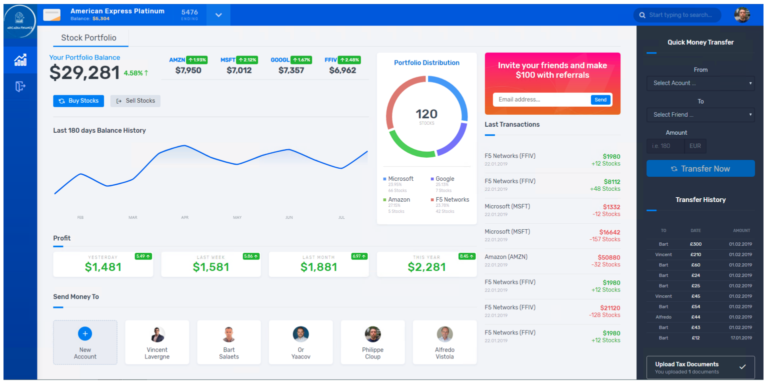


Video of this module :

1.2.3 step 3 - DevOps deploy Refer Friends Application

In this module, we will deploy the Refer Friends container for Arcadia Bank application and we will publish it.

Note: At the end of this module, Arcadia Bank application will look like this.



Video of this module :

1.2.4 step 4 - Protect Arcadia Application

In this module, we will deploy a WAF policy to protect Arcadia Bank application and we will publish it.

Note: We use the new v15.1 Declarative WAF policy. You can retrieve the JSON Policy in the GitLab repo and below.

```
{
    "policy": {
        "name": "policy-fund-1",
        "description": "Policy Example - Rapid Deployment",
        "template": {
            "name": "POLICY_TEMPLATE_RAPID_DEPLOYMENT"
        },
        "enforcementMode": "blocking",
        "server-technologies": [
            {
                "serverTechnologyName": "MySQL"
            },
            {
                "serverTechnologyName": "Unix/Linux"
            },
            {
                "serverTechnologyName": "MongoDB"
            }
        ],
        "signature-settings": {
            "signatureStaging": false
        },
        "policy-builder": {
            "learnOnlyFromNonBotTraffic": false
        }
    }
}
```

Video of this module :