DevOps Lab

CLOUD COMPUTE - GCP

NETWORKING

Home tasks

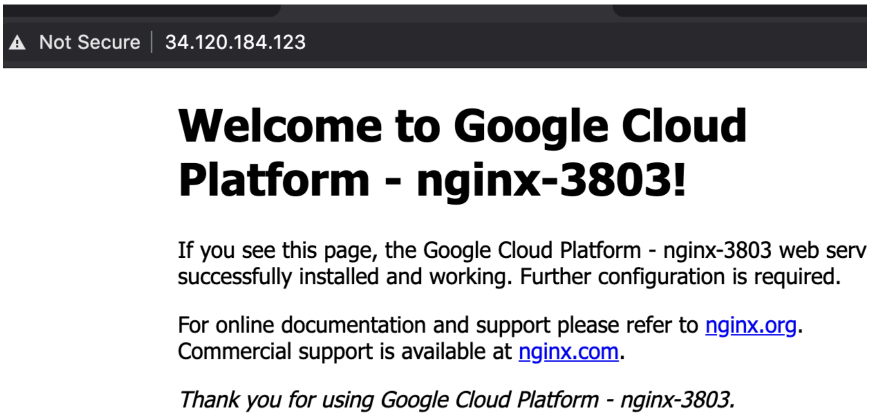
It’s aiming to gain knowledge about Networking in Google Cloud.

TASK 1

Learn about two types of load balancers in Google Cloud Platform:

- a L3 Network Load Balancer and
- a L7 HTTP(s) Load Balancer.

Lab Link: codelabs: LoadBalancers

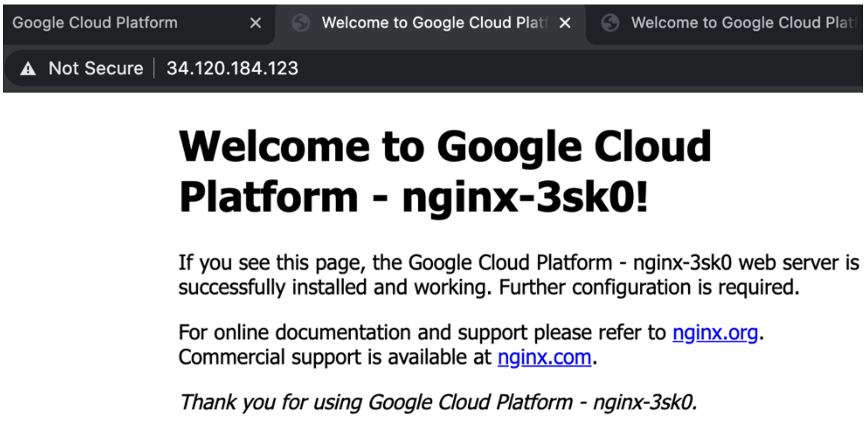


Welcome to Google Cloud Platform - nginx-3803!

If you see this page, the Google Cloud Platform - nginx-3803 web serv successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using Google Cloud Platform - nginx-3803.



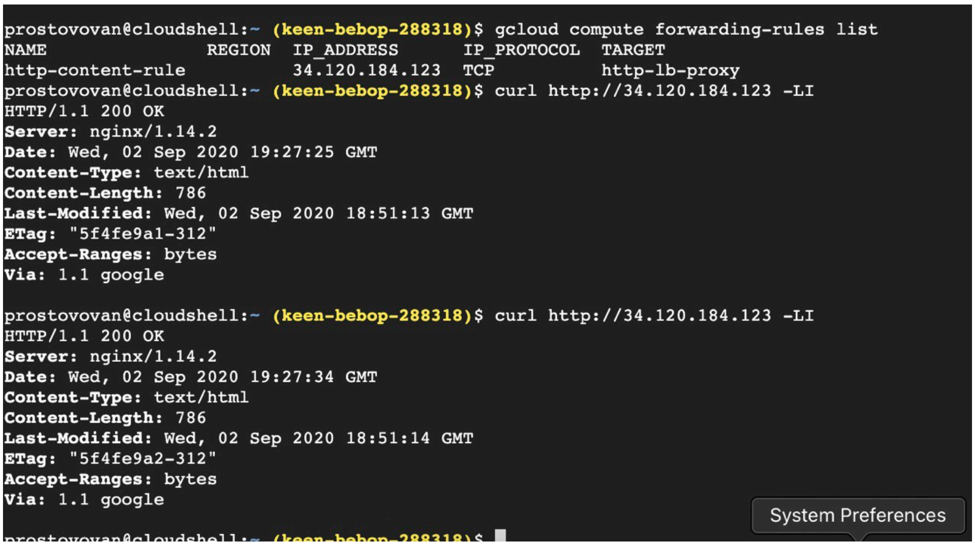
Google Cloud Platform ×
Welcome to Google Cloud Plat ×
Welcome to Google Cloud Plat
Not Secure | 34.120.184.123

Welcome to Google Cloud Platform - nginx-3sk0!

If you see this page, the Google Cloud Platform - nginx-3sk0 web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using Google Cloud Platform - nginx-3sk0.



```
prostovovan@cloudshell:~ (keen-bebop-288318)$ gcloud compute forwarding-rules list
NAME                REGION  IP_ADDRESS      IP_PROTOCOL  TARGET
http-content-rule           34.120.184.123  TCP          http-lb-proxy
prostovovan@cloudshell:~ (keen-bebop-288318)$ curl http://34.120.184.123 -LI
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Wed, 02 Sep 2020 19:27:25 GMT
Content-Type: text/html
Content-Length: 786
Last-Modified: Wed, 02 Sep 2020 18:51:13 GMT
ETag: "5f4fe9a1-312"
Accept-Ranges: bytes
Via: 1.1 google

prostovovan@cloudshell:~ (keen-bebop-288318)$ curl http://34.120.184.123 -LI
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Wed, 02 Sep 2020 19:27:34 GMT
Content-Type: text/html
Content-Length: 786
Last-Modified: Wed, 02 Sep 2020 18:51:14 GMT
ETag: "5f4fe9a2-312"
Accept-Ranges: bytes
Via: 1.1 google

prostovovan@cloudshell:~ (keen-bebop-288318)$
```

TASK 2

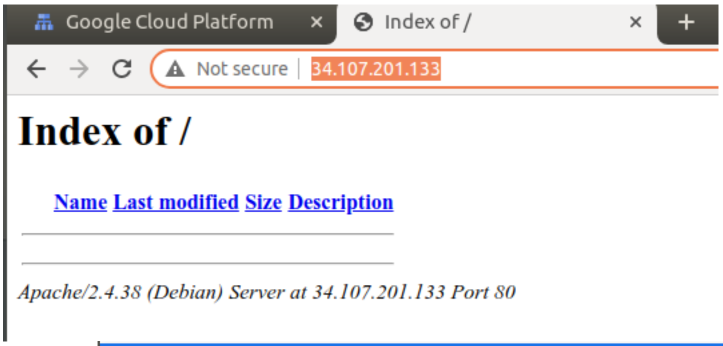
The Objectives are to learn:

- How to measure latency between Google Compute Engine regions and zones
- How to test network connectivity and performance using open source tools
- How to set up up basic firewalling to secure your networks
- How to set up a global HTTP Load Balancer with Managed Instance Groups to automatically scale your resources up and down based on request load
- How to test and monitor your HTTP Load Balancer setup

These exercises are ordered to reflect a common cloud developer experience as follows:

1. Set up your lab environment and learn how to work with your GCP environment.
2. Use of common open source tools to explore your network around the world.
3. Deploy a common use case: use of HTTP Load Balancing and Managed Instance Groups to host a scalable, multi-region web server.
4. Testing and monitoring your network and instances.
5. Cleanup.

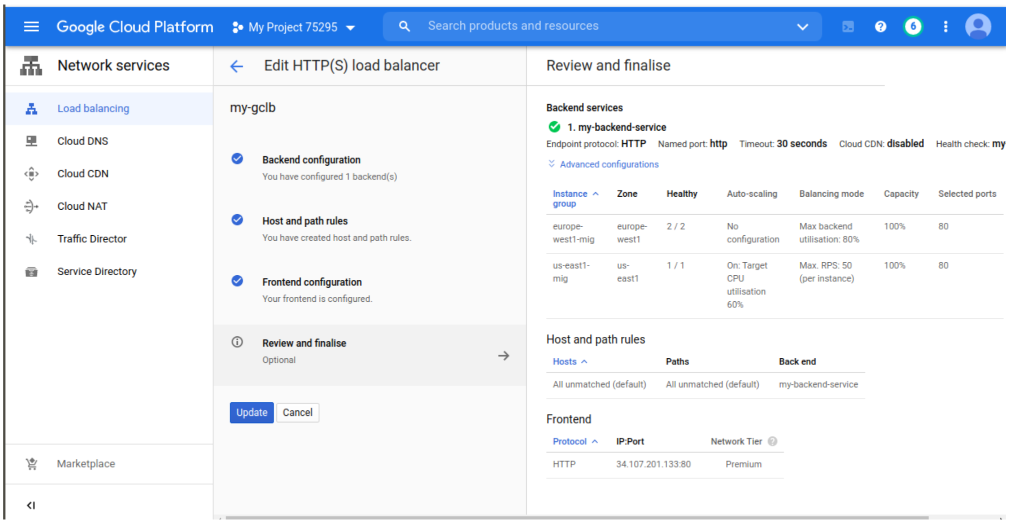
Lab Link: codelabs: Neworking 101



Index of /

Name Last modified Size Description

Apache/2.4.38 (Debian) Server at 34.107.201.133 Port 80



my-gclb

Backend configuration
You have configured 1 backend(s)

Host and path rules
You have created host and path rules.

Frontend configuration
Your frontend is configured.

Review and finalise
Optional

Review and finalise

Backend services

1. my-backend-service

Endpoint protocol: **HTTP** Named port: **http** Timeout: **30 seconds** Cloud CDN: **disabled** Health check: **my**

Advanced configurations

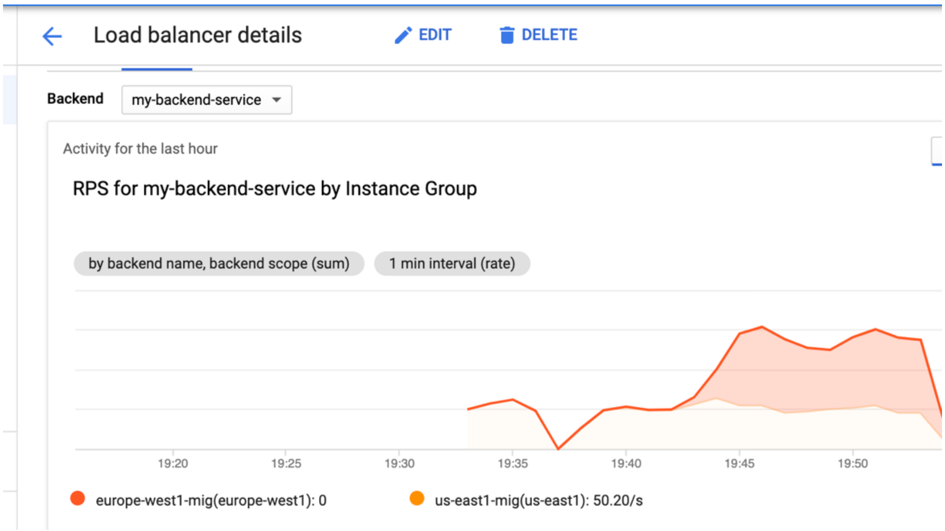
Instance group	Zone	Healthy	Auto-scaling	Balancing mode	Capacity	Selected ports
europe-west1-mig	europe-west1	2 / 2	No configuration	Max backend utilisation: 80%	100%	80
us-east1-mig	us-east1	1 / 1	On: Target CPU utilisation 60%	Max. RPS: 50 (per instance)	100%	80

Host and path rules

Hosts	Paths	Back end
All unmatched (default)	All unmatched (default)	my-backend-service

Frontend

Protocol	IP:Port	Network Tier
HTTP	34.107.201.133:80	Premium

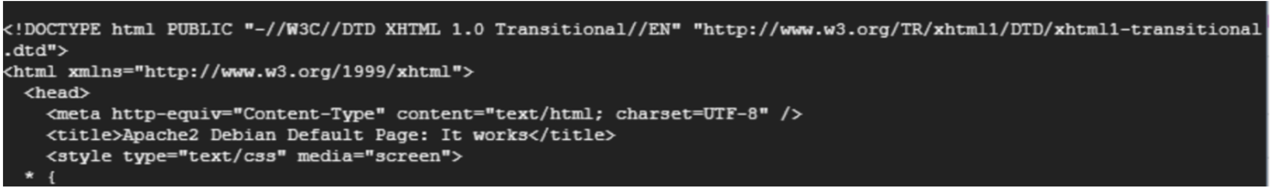

Load balancer details
EDIT
DELETE
Backend
my-backend-service
Activity for the last hour
RPS for my-backend-service by Instance Group
by backend name, backend scope (sum)
1 min interval (rate)
19:20
19:25
19:30
19:35
19:40
19:45
19:50
● europe-west1-mig(europe-west1): 0
● us-east1-mig(us-east1): 50.20/s

TASK 3

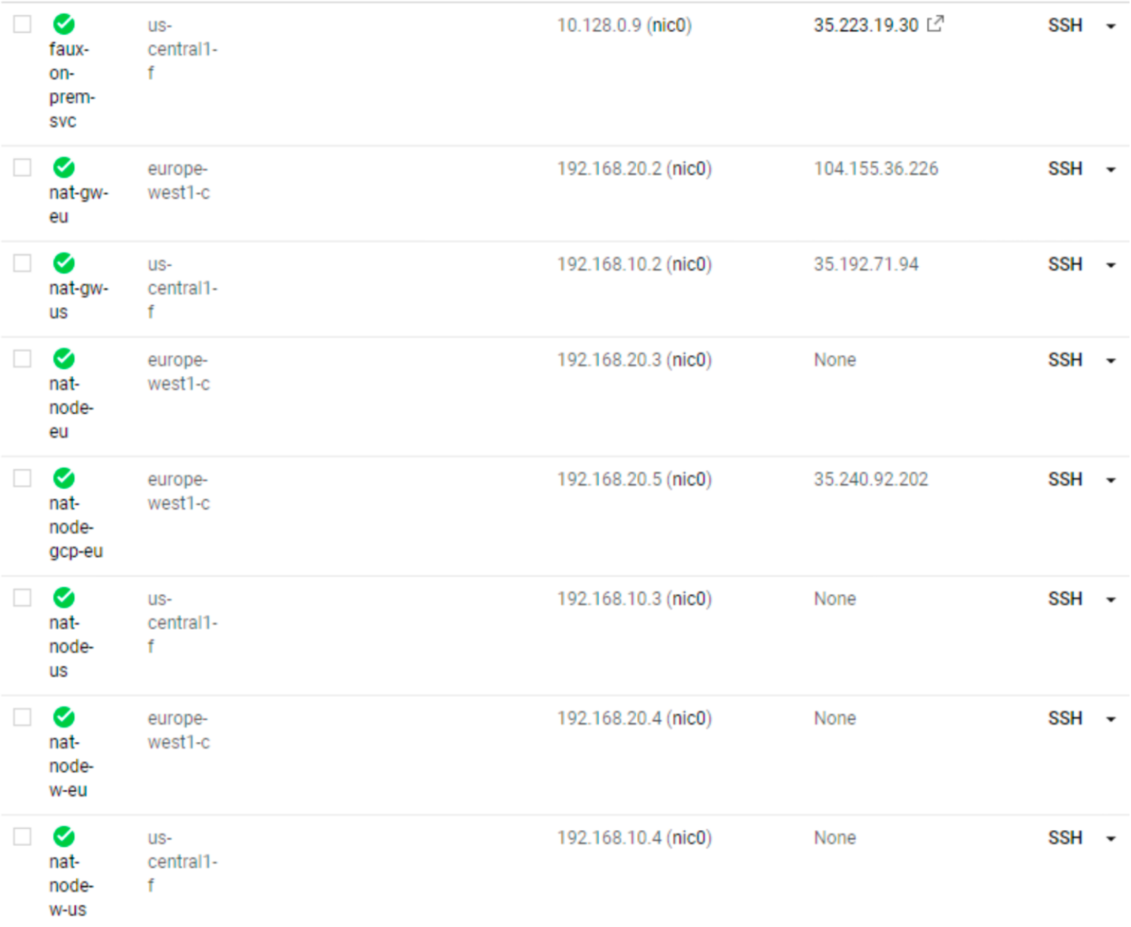
The Objectives are to learn:

- Setting up NAT gateways
- How to restrict network traffic that certain tiers of an app cannot talk to each other
- Setting up alternate connectivity options to instances
- Map an external service to look like an internal service
- How to setup an Egress proxy limiting access to specific resources

Lab Link: codelabs: Neworking 102



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Debian Default Page: It works</title>
    <style type="text/css" media="screen">
  * {
```



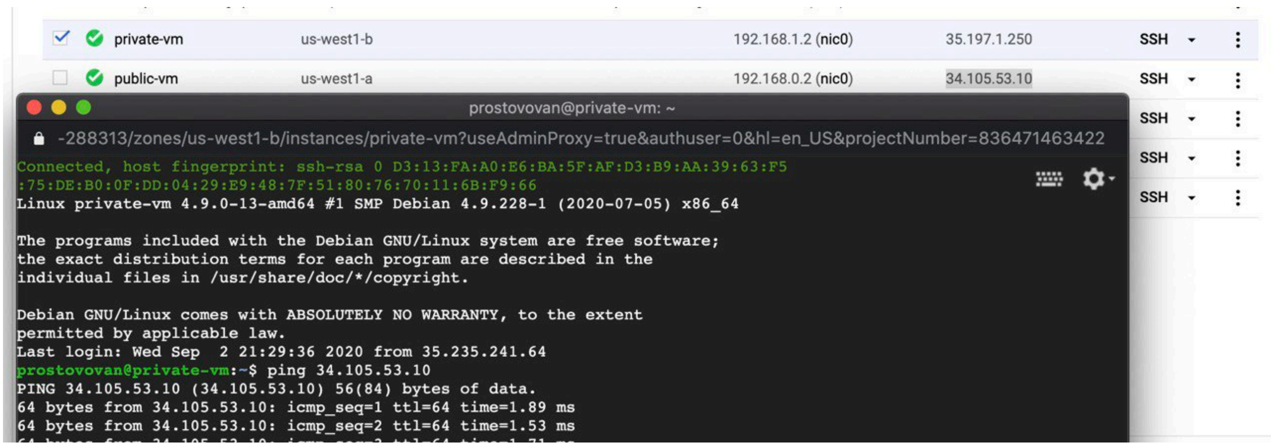
□	✓	faux-on-prem-svc	us-central1-f	10.128.0.9 (nic0)	35.223.19.30	SSH ▾
□	✓	nat-gw-eu	europe-west1-c	192.168.20.2 (nic0)	104.155.36.226	SSH ▾
□	✓	nat-gw-us	us-central1-f	192.168.10.2 (nic0)	35.192.71.94	SSH ▾
□	✓	nat-node-eu	europe-west1-c	192.168.20.3 (nic0)	None	SSH ▾
□	✓	nat-node-gcp-eu	europe-west1-c	192.168.20.5 (nic0)	35.240.92.202	SSH ▾
□	✓	nat-node-us	us-central1-f	192.168.10.3 (nic0)	None	SSH ▾
□	✓	nat-node-w-eu	europe-west1-c	192.168.20.4 (nic0)	None	SSH ▾
□	✓	nat-node-w-us	us-central1-f	192.168.10.4 (nic0)	None	SSH ▾

TASK 4

The Objectives are to learn:

- Secure app in custom network

Lab Link: codelabs: custom_network



☑	✓	private-vm	us-west1-b	192.168.1.2 (nic0)	35.197.1.250	SSH ▾	⋮
□	✓	public-vm	us-west1-a	192.168.0.2 (nic0)	34.105.53.10	SSH ▾	⋮

prostovovan@private-vm: ~
-288313/zones/us-west1-b/instances/private-vm?useAdminProxy=true&authuser=0&hl=en_US&projectNumber=836471463422

```
Connected, host fingerprint: ssh-rsa 0 D3:13:FA:A0:E6:BA:5F:AF:D3:B9:AA:39:63:F5
:75:DE:B0:0F:DD:04:29:E9:48:7F:51:80:76:70:11:6B:F9:66
Linux private-vm 4.9.0-13-amd64 #1 SMP Debian 4.9.228-1 (2020-07-05) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Sep  2 21:29:36 2020 from 35.235.241.64
prostovovan@private-vm:~$ ping 34.105.53.10
PING 34.105.53.10 (34.105.53.10) 56(84) bytes of data.
64 bytes from 34.105.53.10: icmp_seq=1 ttl=64 time=1.89 ms
64 bytes from 34.105.53.10: icmp_seq=2 ttl=64 time=1.53 ms
```

TASK 5

Create network configuration via terraform.
Resources should be used:

1) **google_compute_network** (to create network)
   https://www.terraform.io/docs/providers/google/r/compute_network.html

   Network name: ${student_name}-vpc

2) **google_compute_firewall**
   (to create rules for external (allow 80,22) /internal access (allow 0-65535))
   https://www.terraform.io/docs/providers/google/r/compute_firewall.html

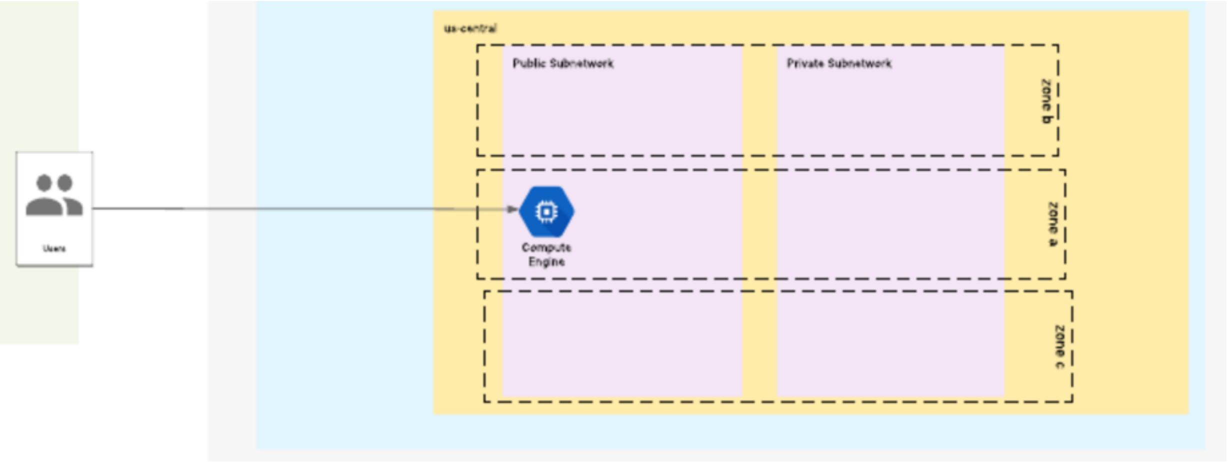
3) **google_compute_subnetwork**
   https://www.terraform.io/docs/providers/google/r/compute_subnetwork.html
   ranges:
   - Public range: 10.”${student_IDnum}”.1.0/24
   - Private range: 10.”${student_IDnum}”.2.0/24

4) **google_compute_instance**
   https://www.terraform.io/docs/providers/google/r/compute_instance.html

   1. nginx with default page “Hello from ${student_name}”

All resources should contain description (where it’s possible)

Network topology.



All **reports**/code please place into repository: https://github.com/MNT-Lab/google-cloud-module into appropriate branches: *first char of name + surname*.

For example:
Student: Siarhei Ivanou
Branch Name: **sivanou**

Format depends on case: README.md/scripts/terraform files

Email pattern: [MNT-CD-8.3]-FirstName-LastName

Email should contain the link to personalized branch.