

LAB: Launch Instances with User-Data

You need:

- An active AWS Account with Root Account Access

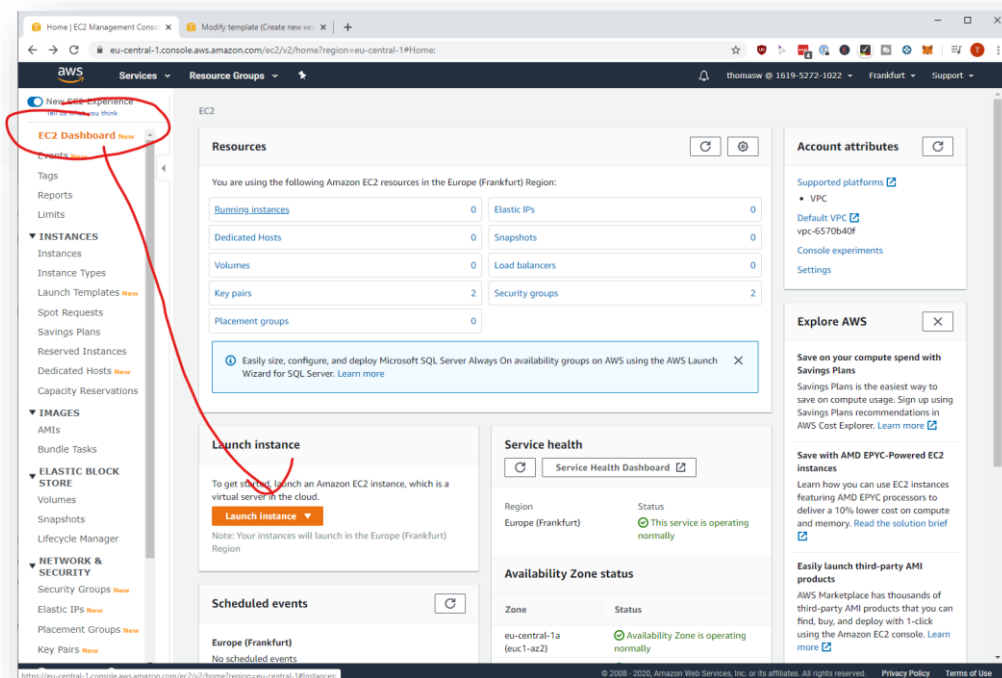
Duration of the Lab: 15 Minutes.

Goal: Understand User-Data

Difficulty: Easy.

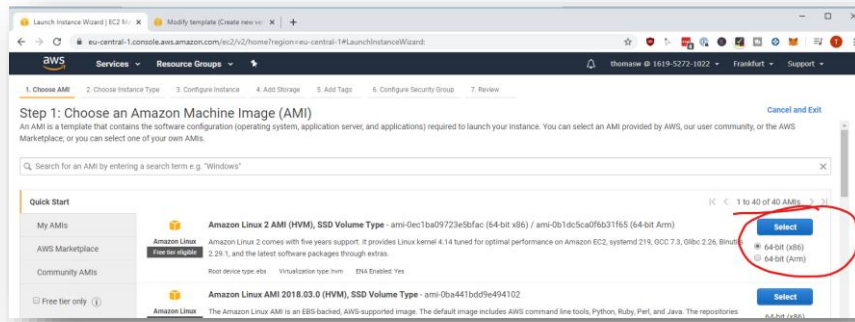
Launch with User-Data

Login into the EC2 Dashboard and Select “Launch Instance”.

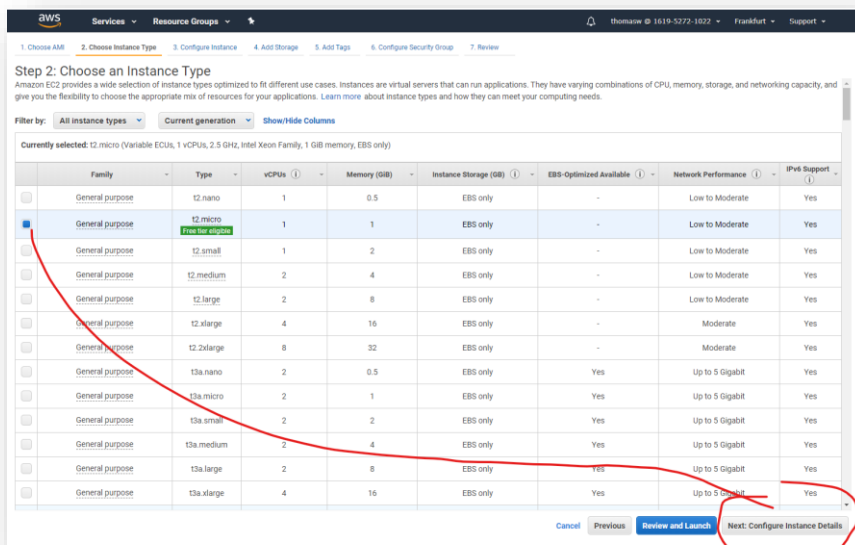


Select an AMI, I selected the Amazon Linux 2 AMI:

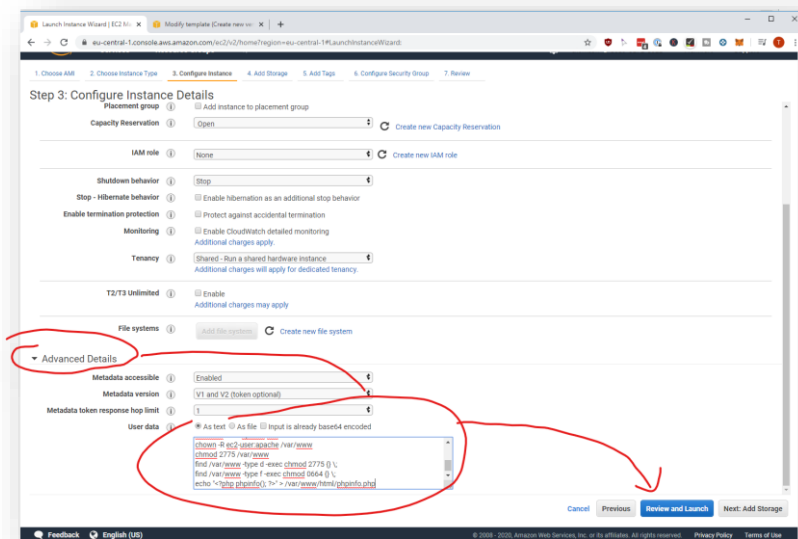
Complete AWS DevOps Training for Beginners



Then choose an Instance Type, like the t2.micro:



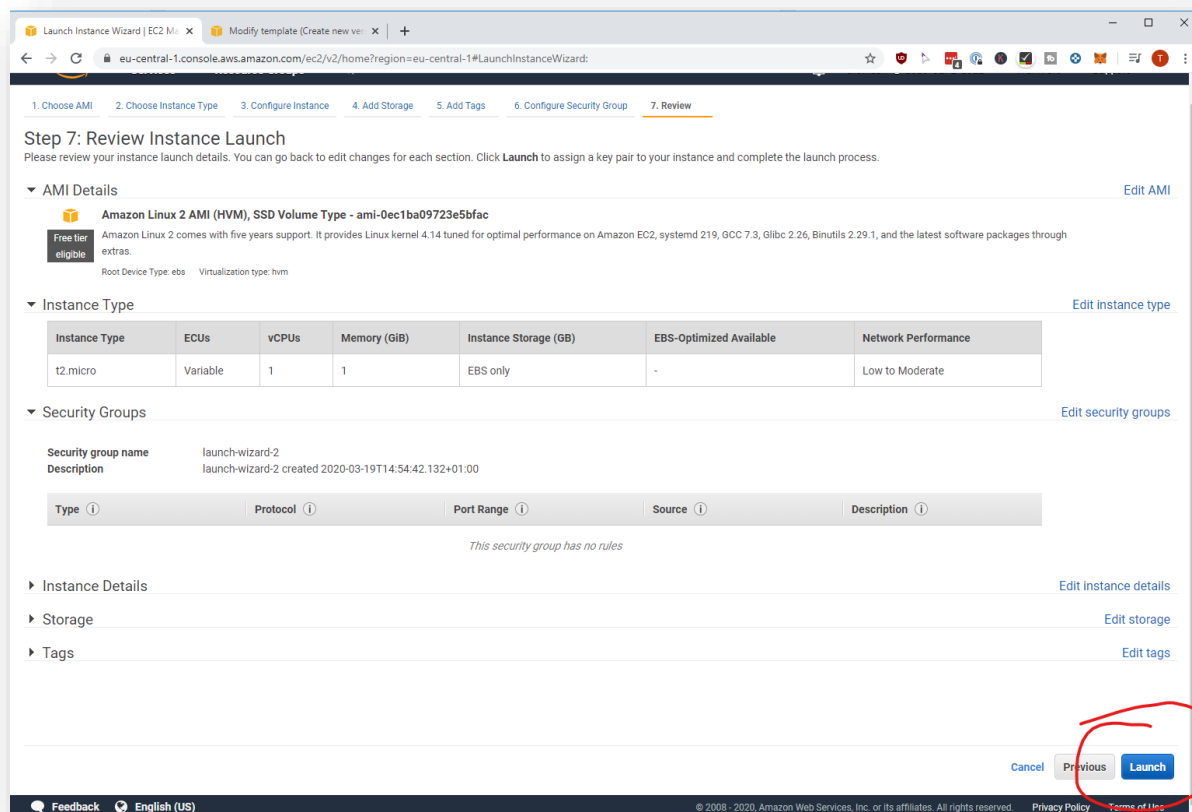
On the next page, scroll down to the Advanced Details. You will find the User data field:



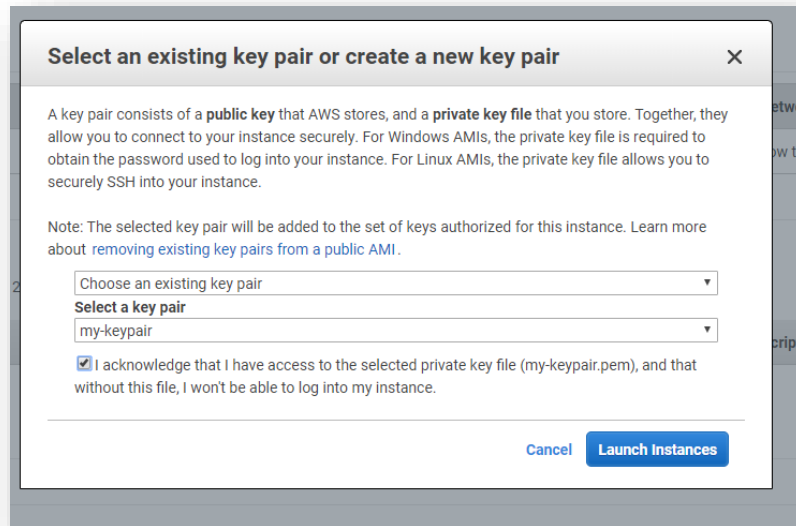
Paste the following script into the user data field and click review and launch.

```
#!/bin/bash
yum update -y
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
yum install -y httpd mariadb-server
systemctl start httpd
systemctl enable httpd
usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} \;
find /var/www -type f -exec chmod 0664 {} \;
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

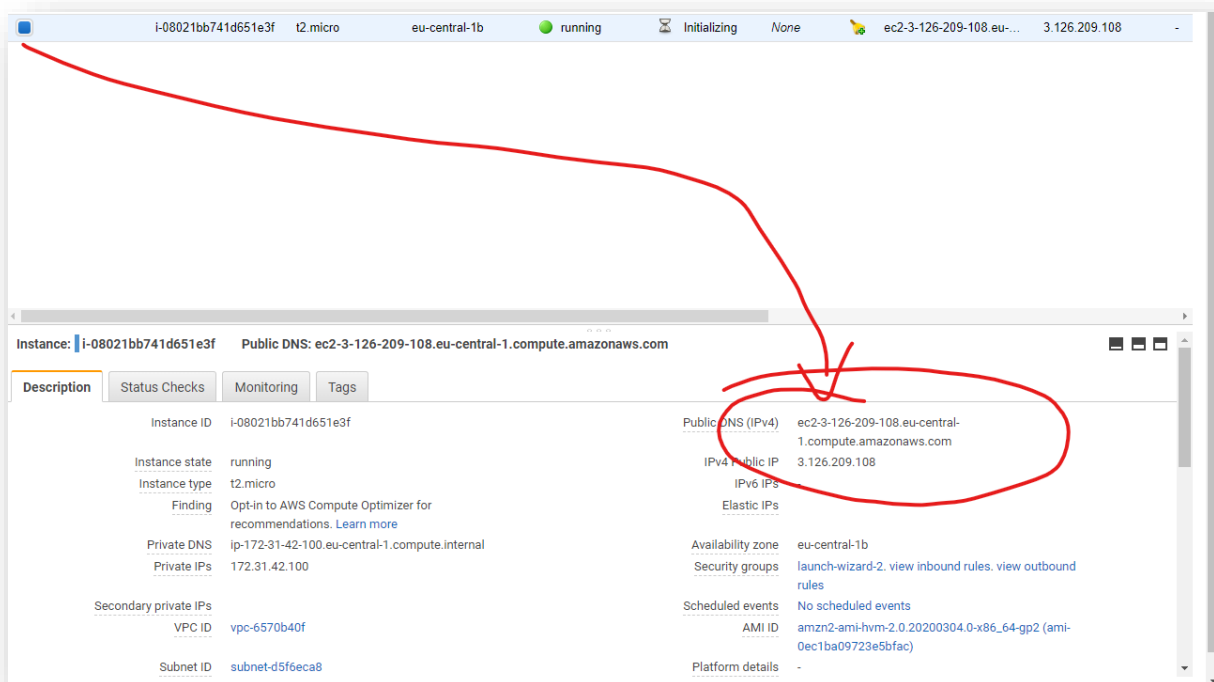
Click Launch:



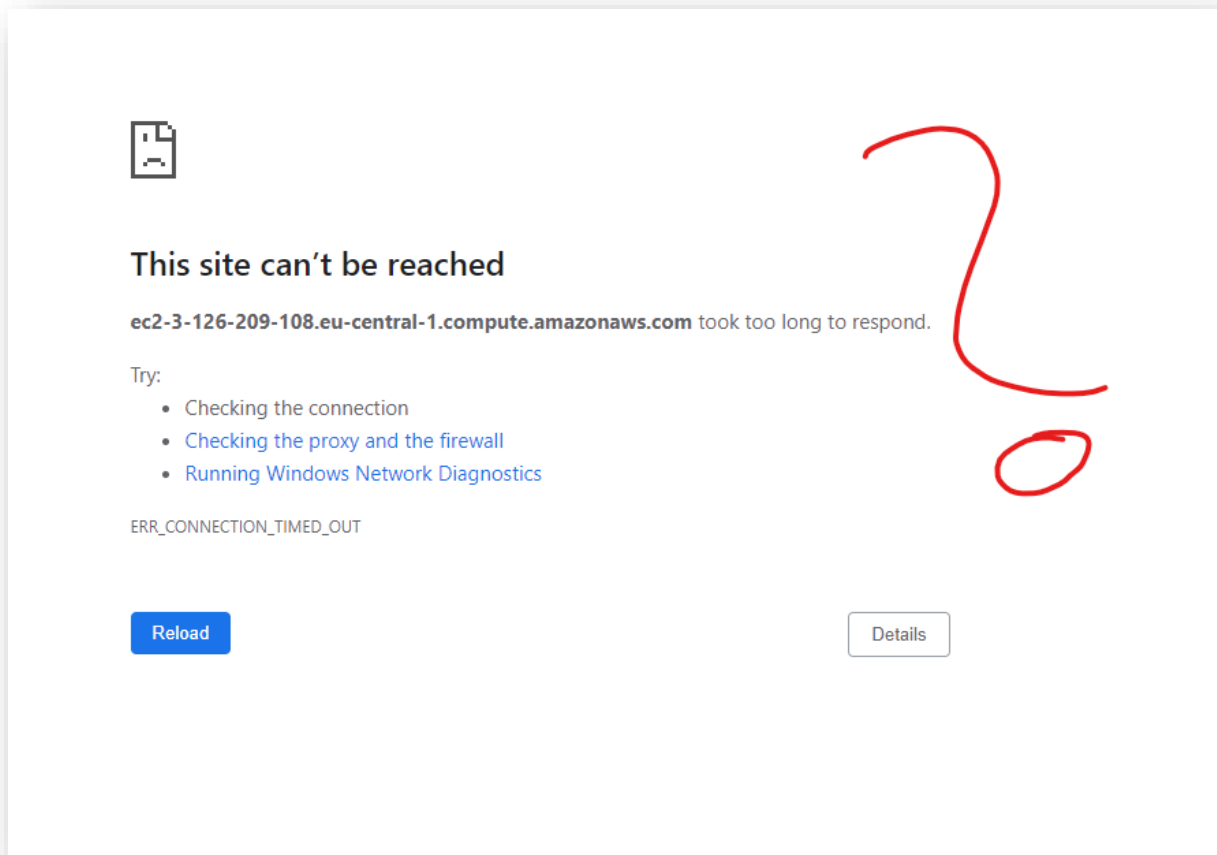
Select your Keypair you downloaded earlier and Launch Instance:



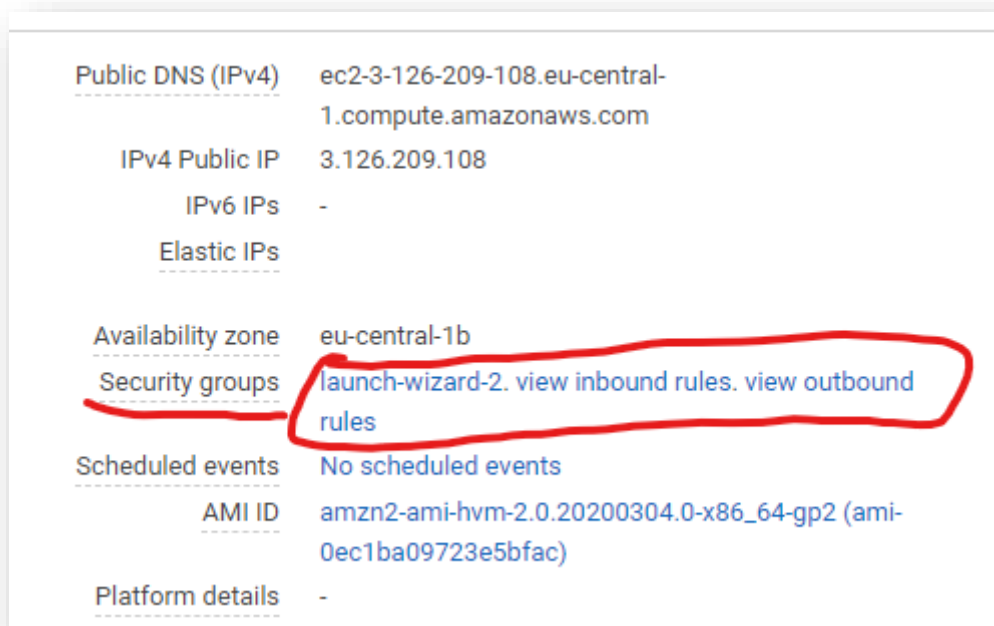
Wait until the Instance is running and copy the public DNS (or IP4)



Open the URL in your browser. What happens here? Why can't the site be reached?

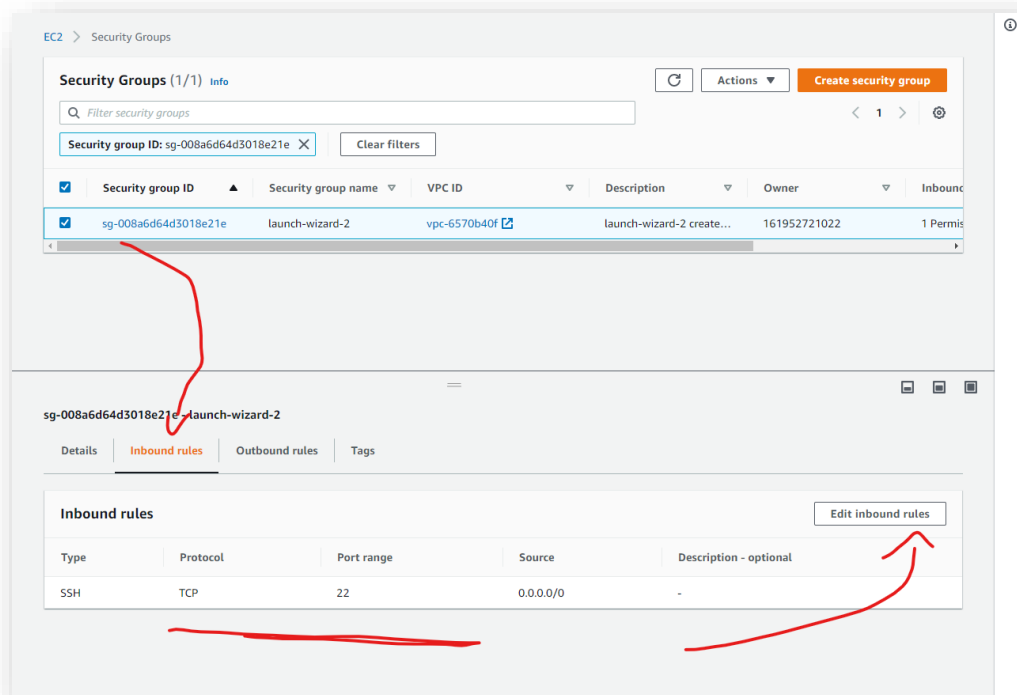


The Security group of your Instance doesn't allow incoming connections on Port 80. We have to extend this. On the instance you see the security groups. Click on "view inbound rules":

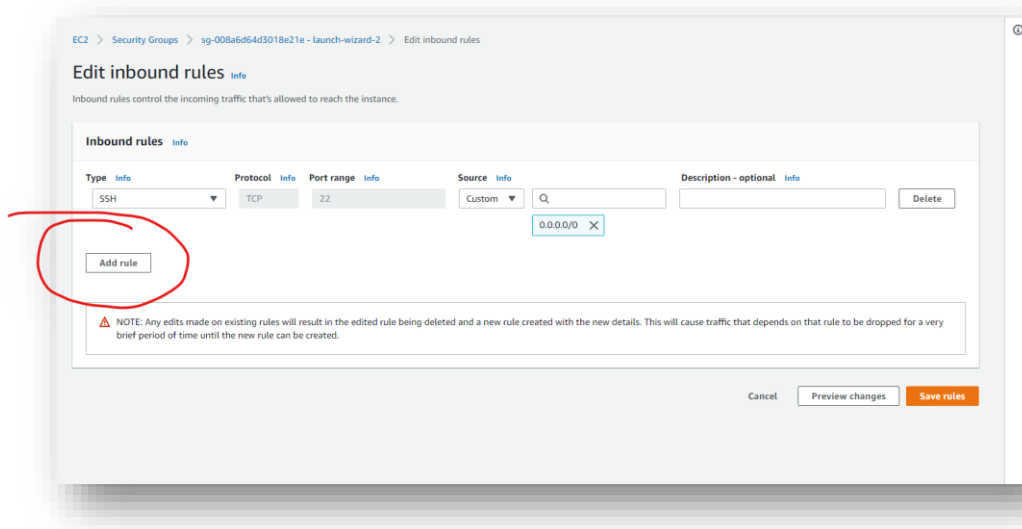


Edit the inbound rules:

Complete AWS DevOps Training for Beginners

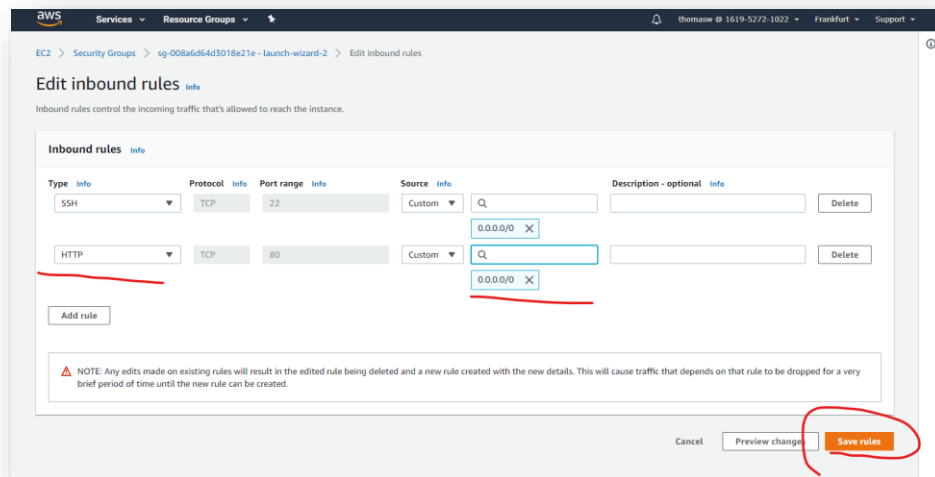


Add a new Rule:

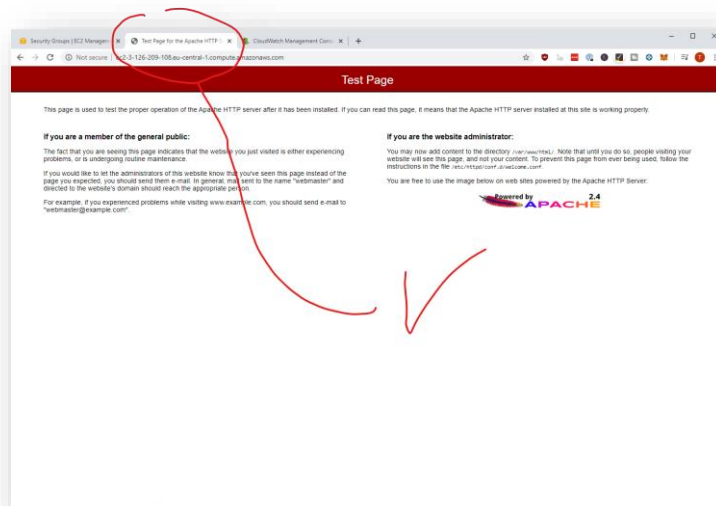


Allow HTTP from Everywhere.

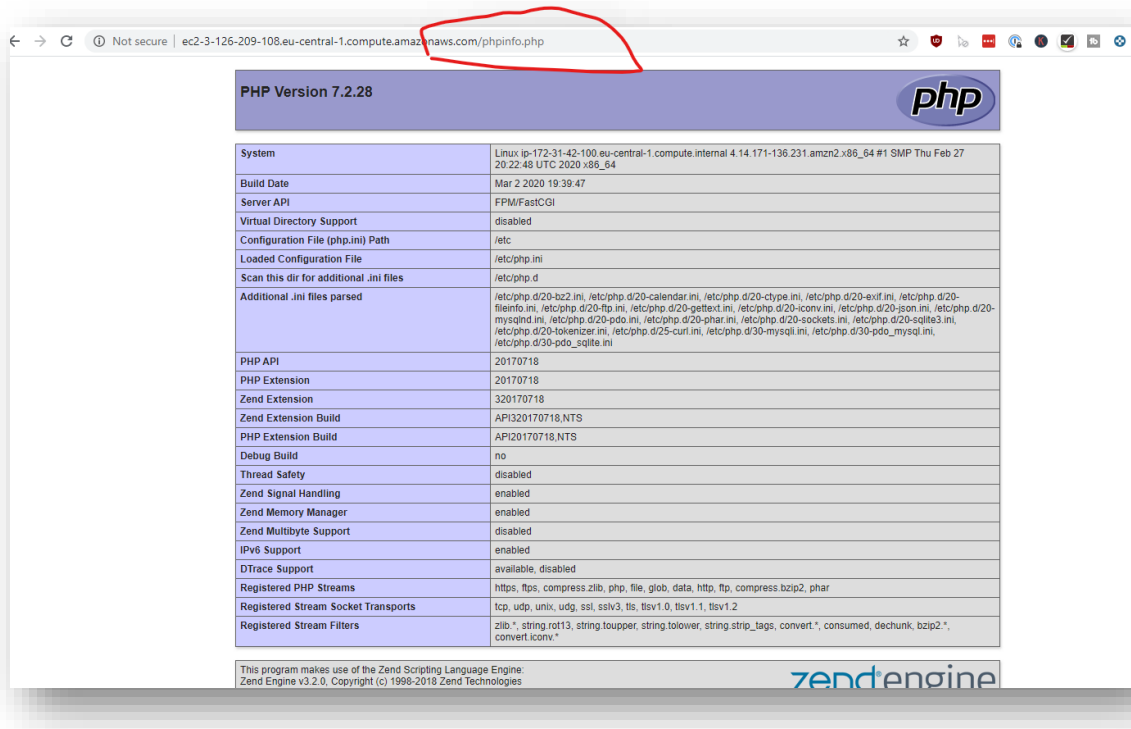
Then hit "Save rules":



Reload the Website for the Instance and see it works now. Apache delivers the Test-Page.



Change the URL to phpinfo.php to see the script which gets injected on startup:



PHP Version 7.2.28

System	Linux ip-172-31-42-100.eu-central-1.compute.internal 4.14.171-136.231.amzn2.x86_64 #1 SMP Thu Feb 27 20:22:48 UTC 2020 x86_64
Build Date	Mar 2 2020 19:39:47
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-json.ini, /etc/php.d/20-mysqli.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/25-curl.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API20170718.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, bzip2.*, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.2.0. Copyright (c) 1998-2018 Zend Technologies

Note: You can stop the instance now if you continue another day, but don't remove it, because we will need it in the next assignment for understanding the meta-data

Lab End
