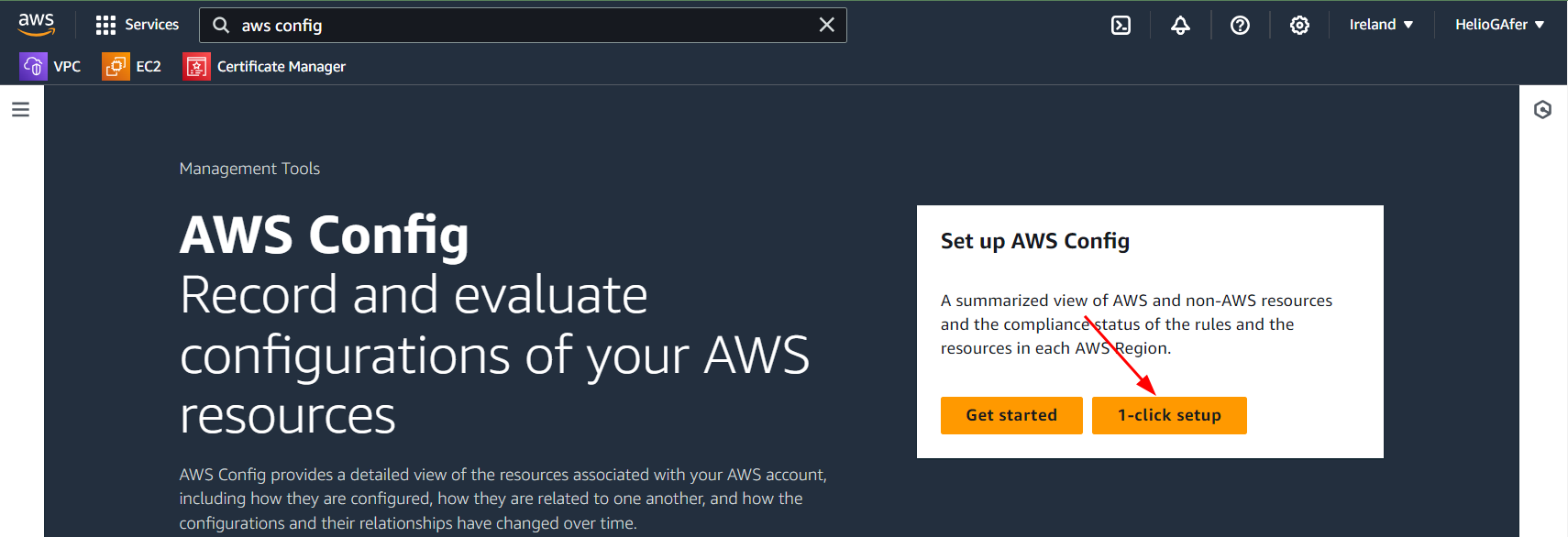
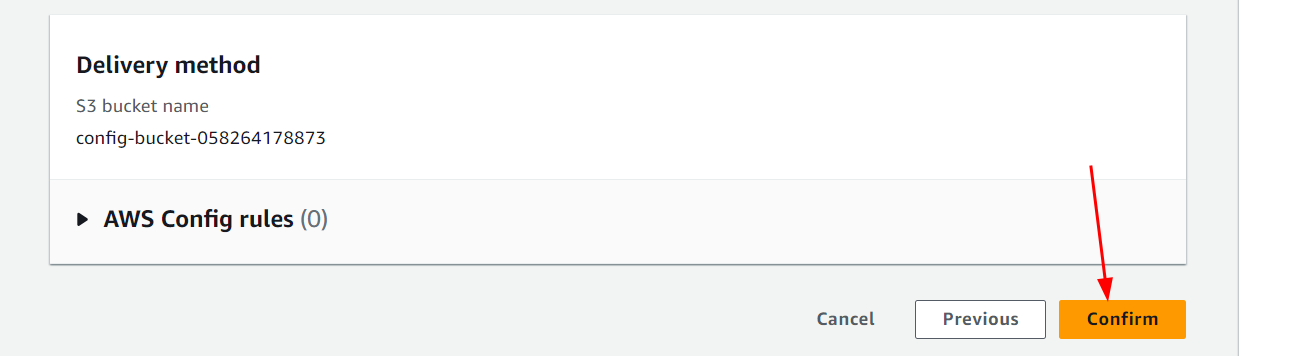
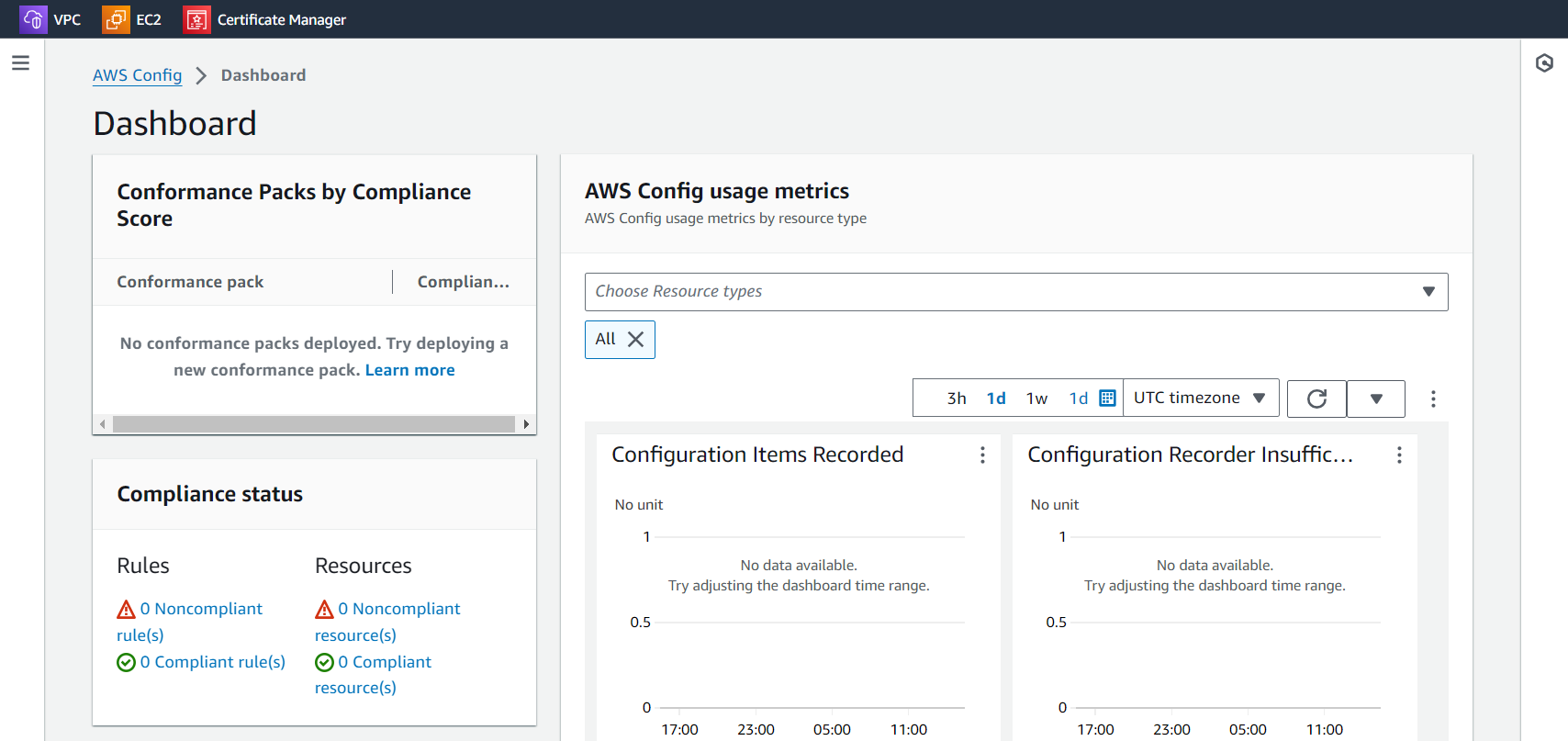
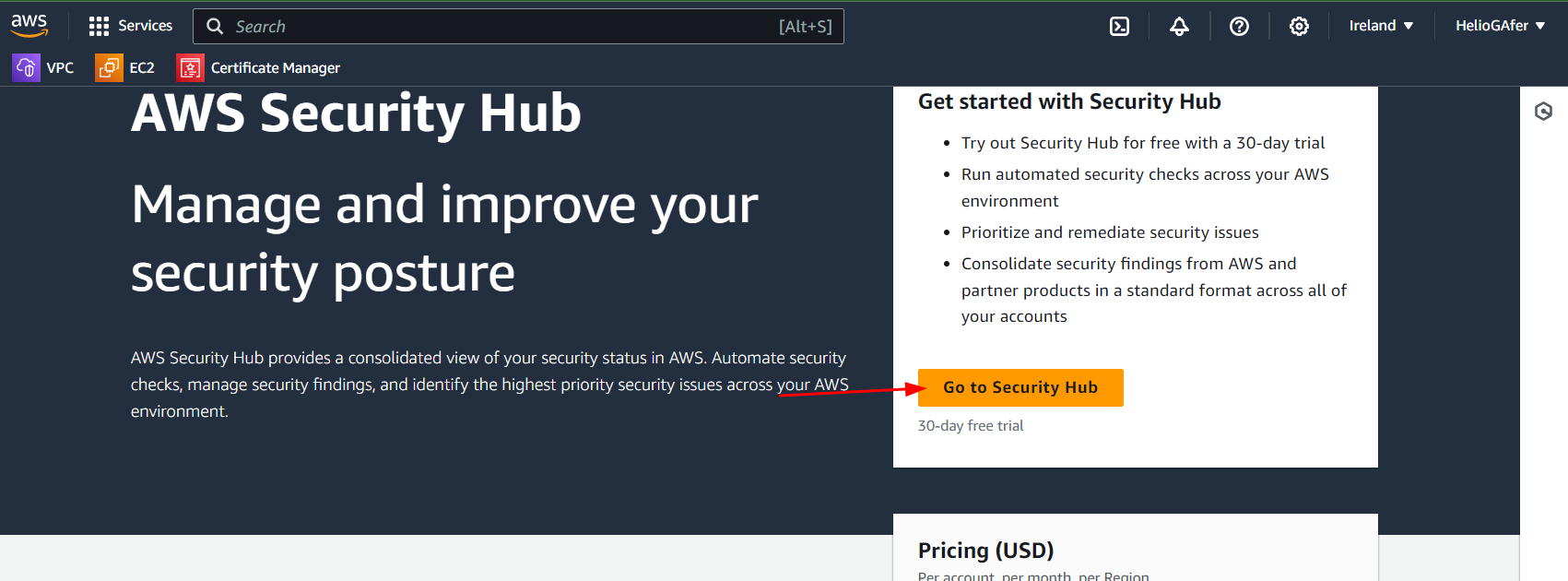
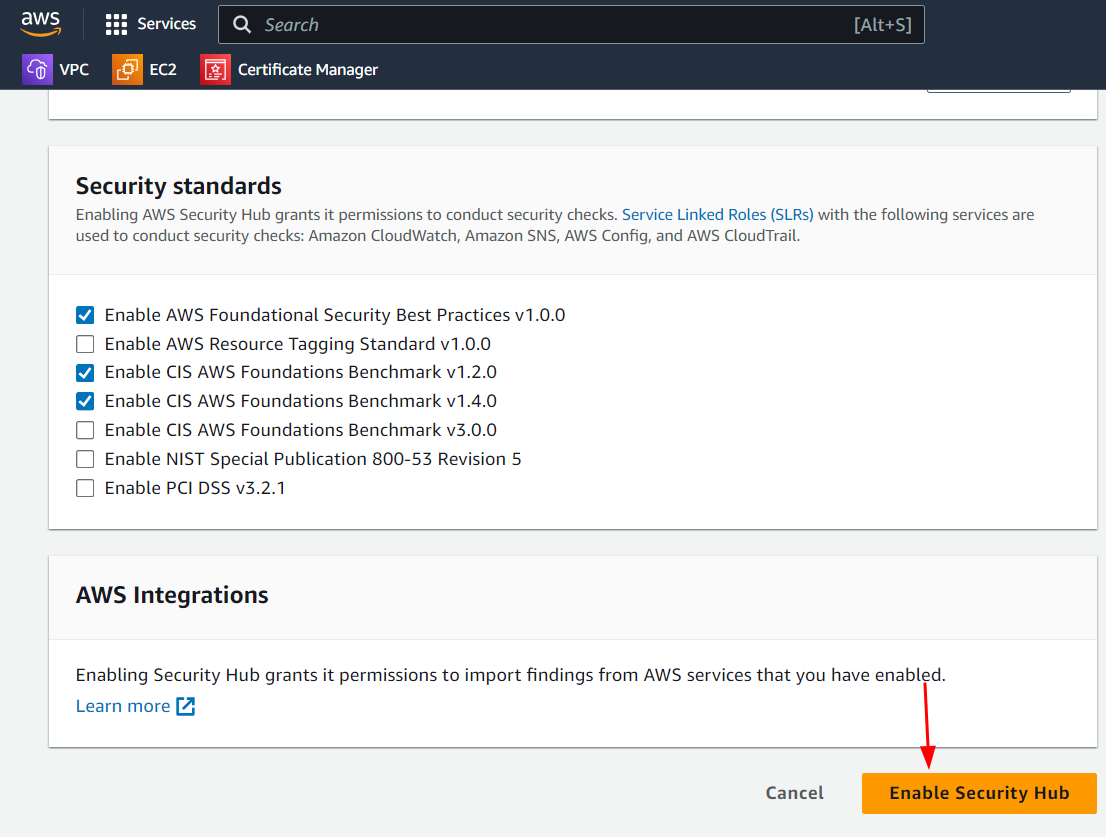
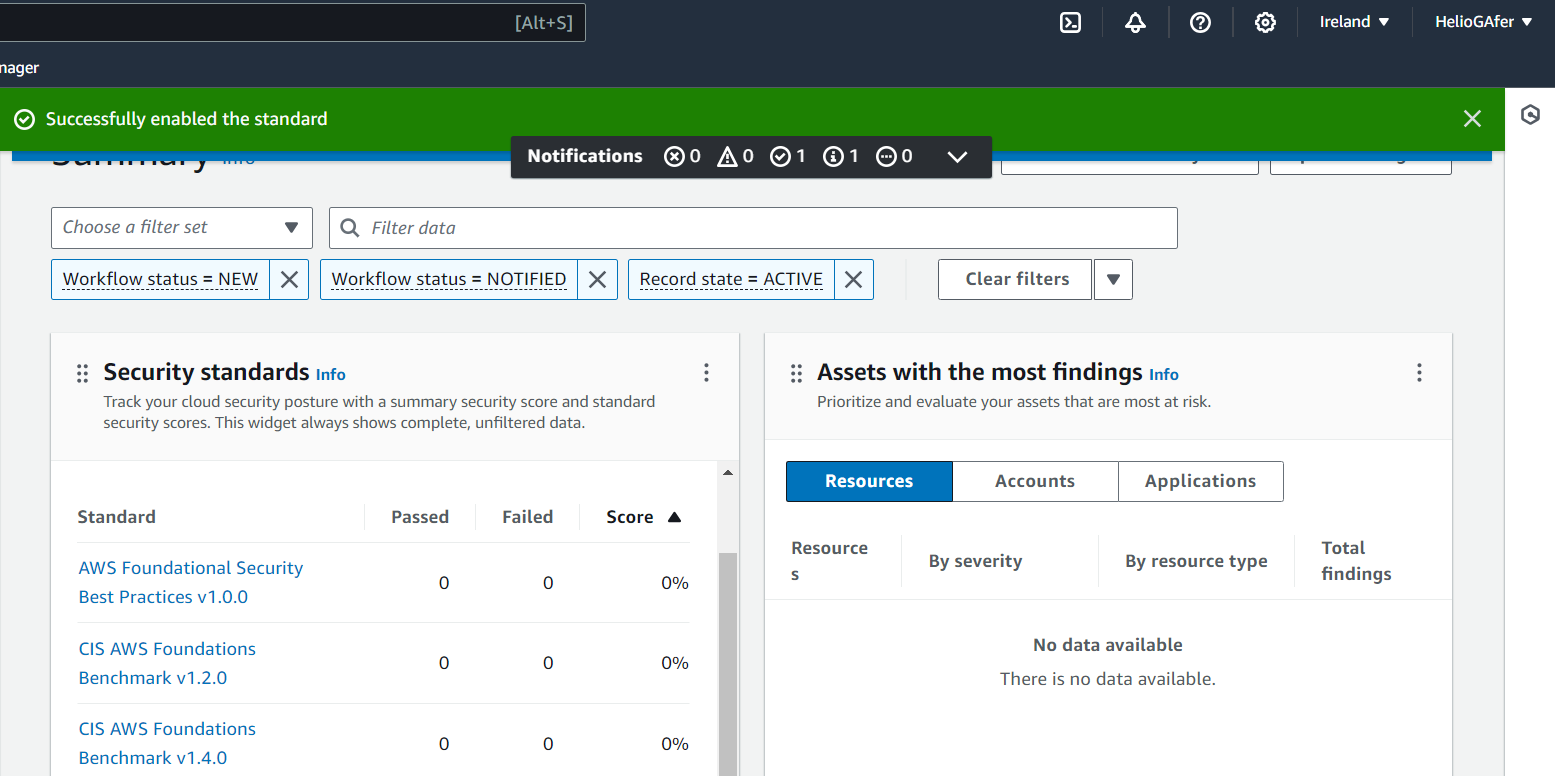
Hélio Ferreira 09/05/2024

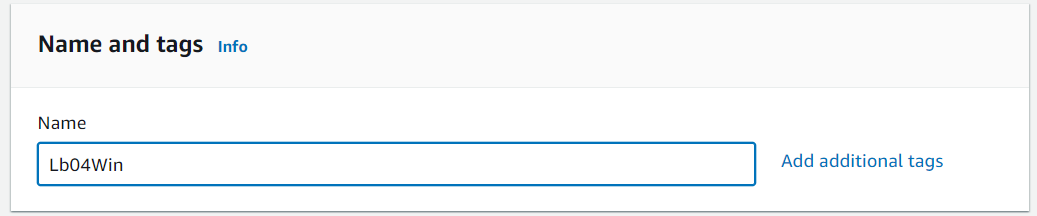
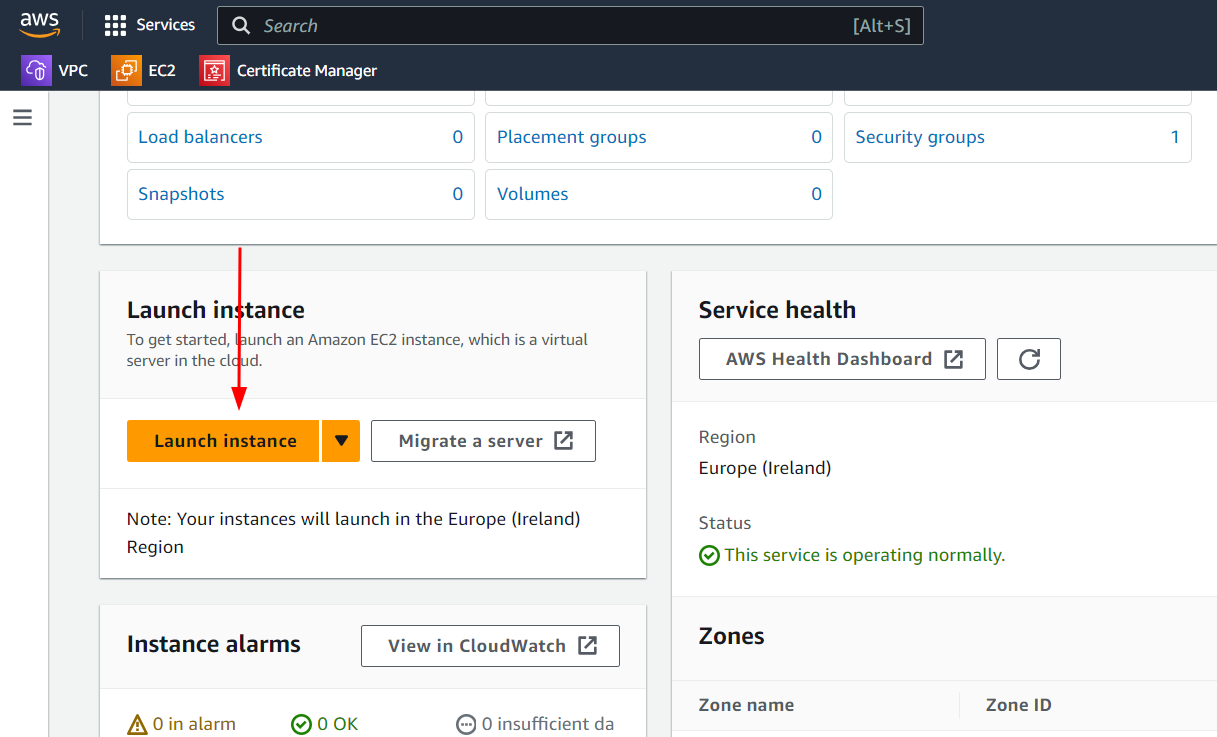
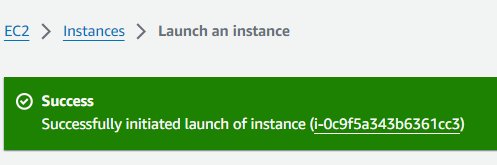
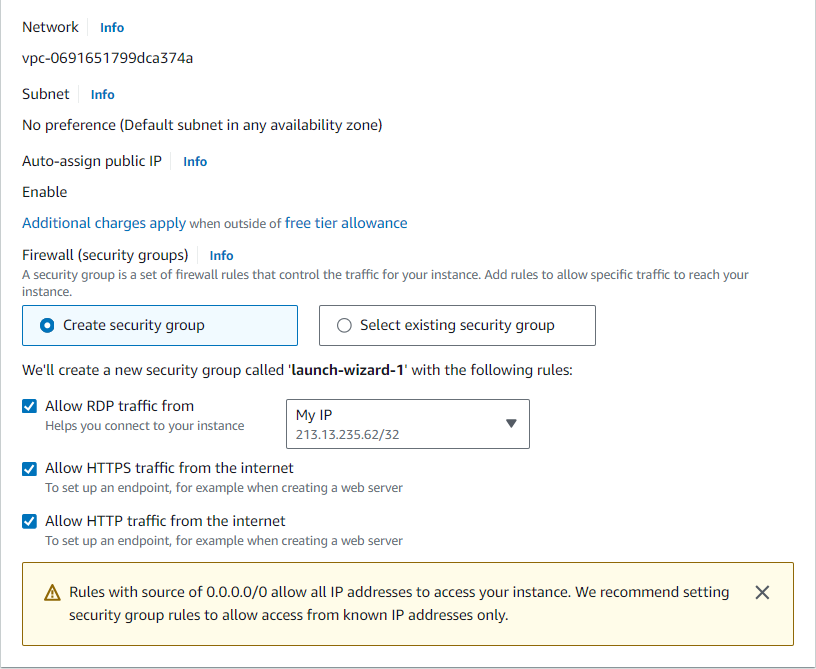
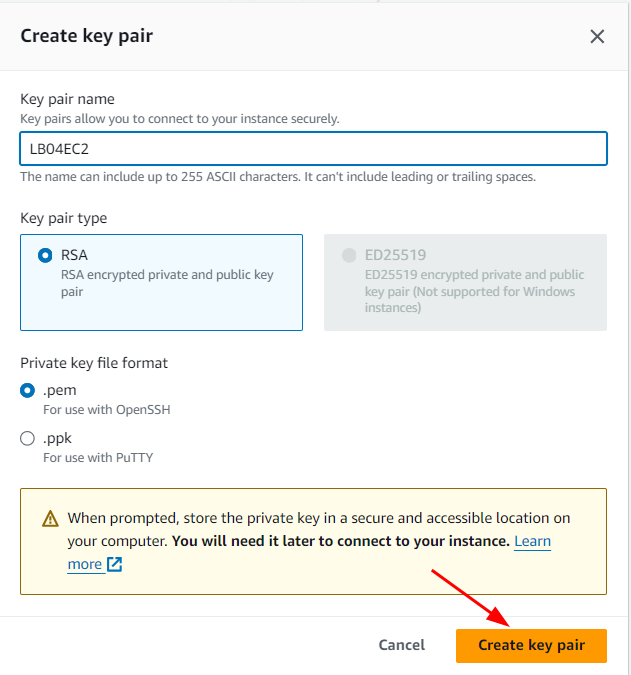
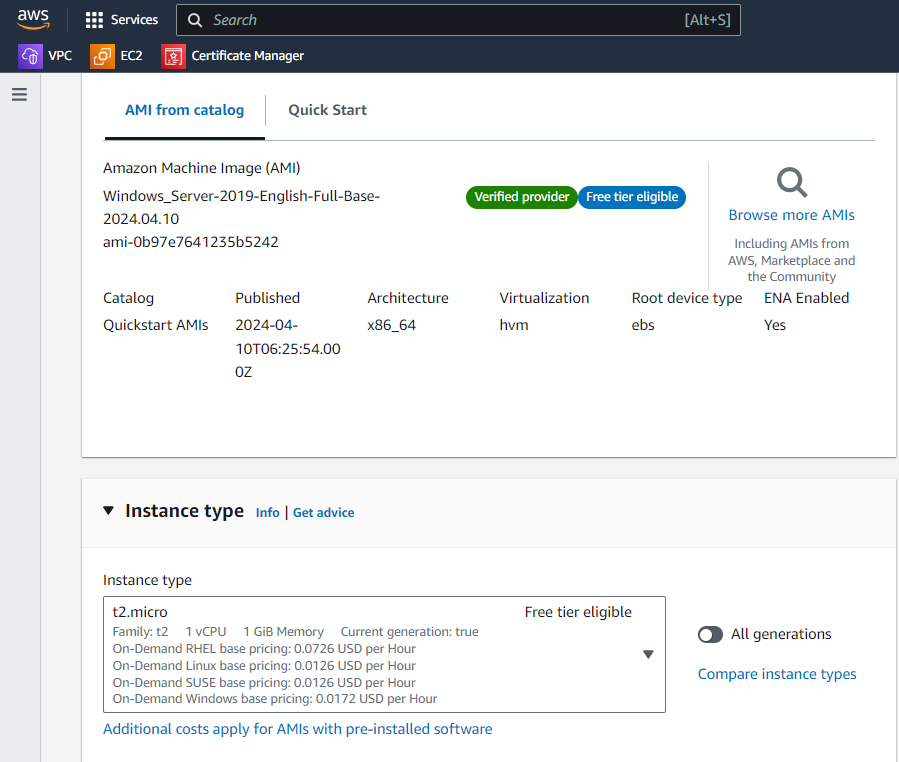
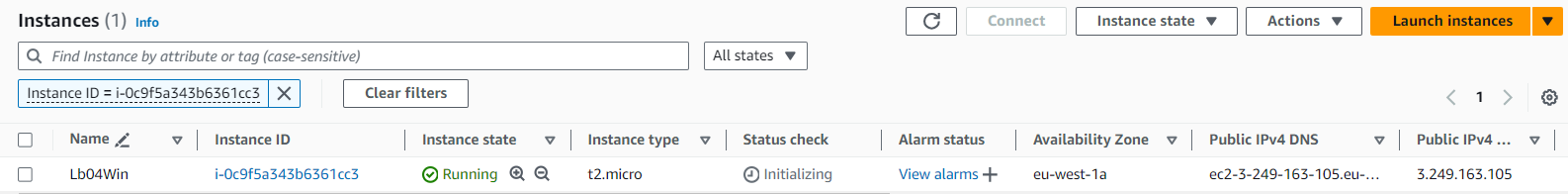
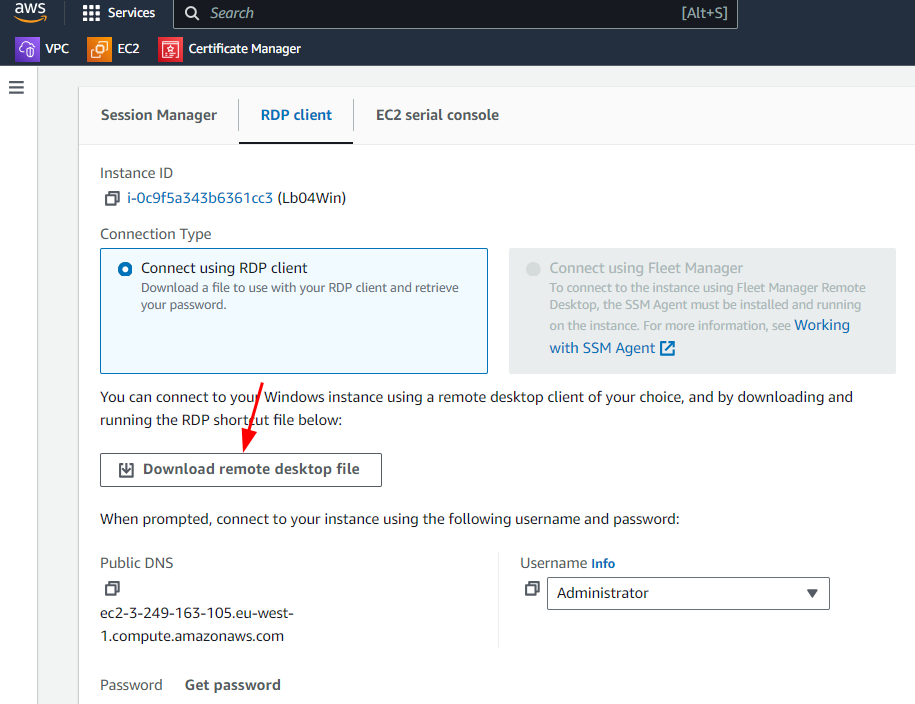
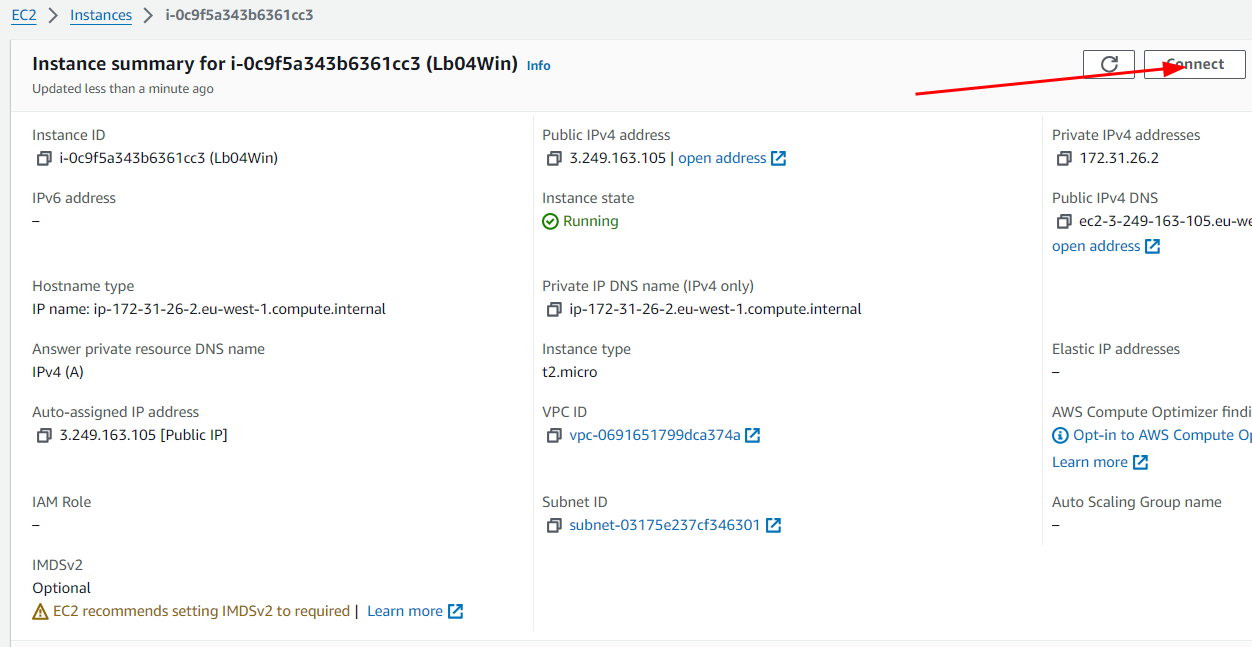
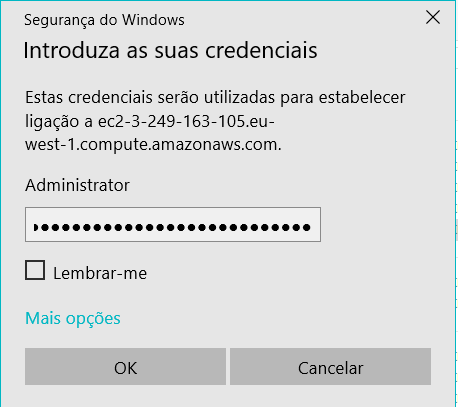
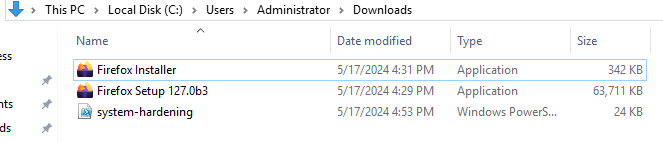
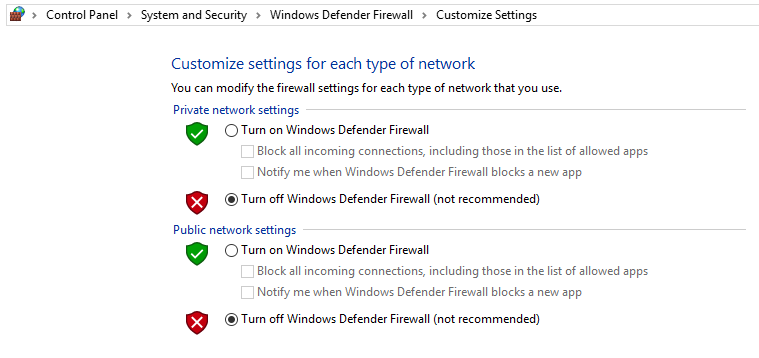
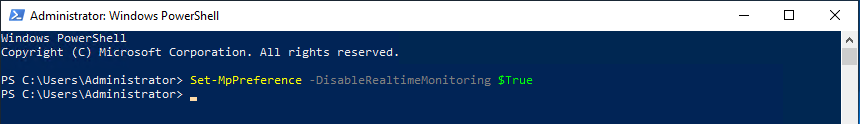
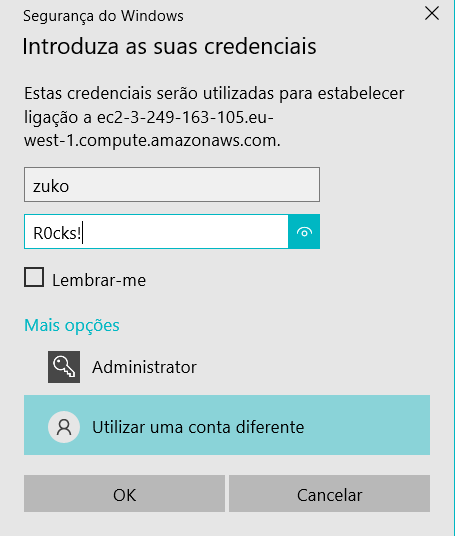
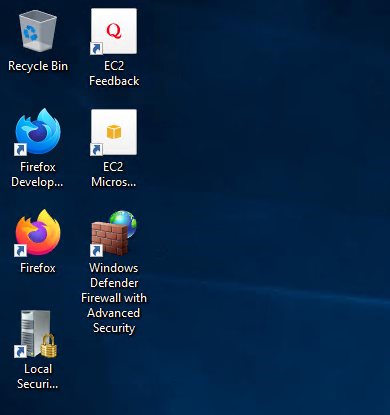
# **Systems Hardening with CIS Standards**

### **Part 1: Staging**

For this lab you’ll need an AWS account with free tier hours available.

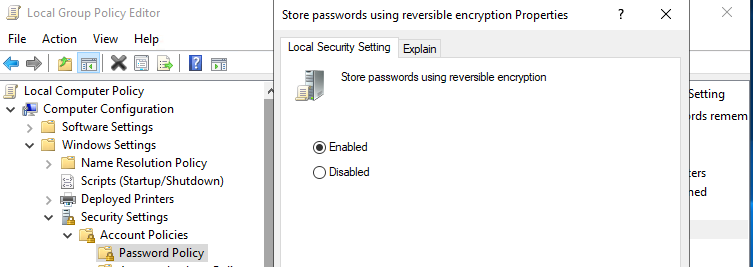
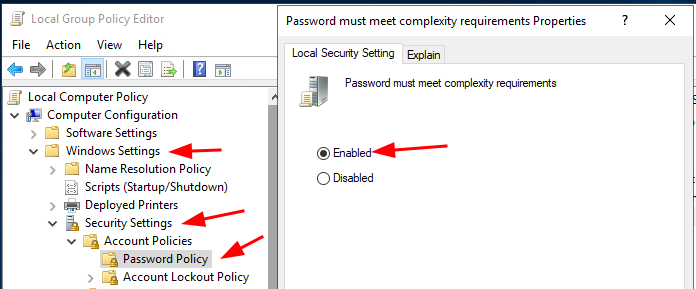
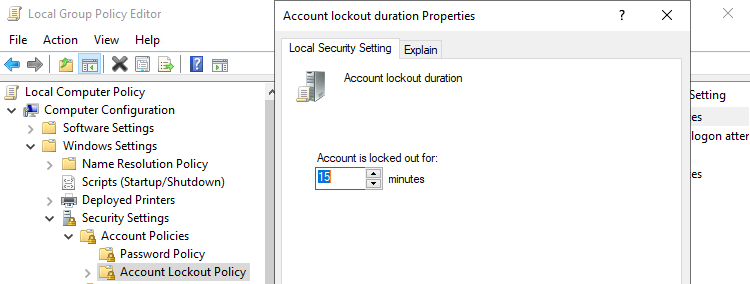
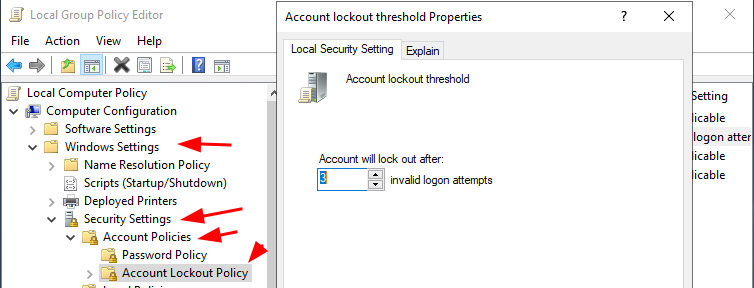
* Enable AWS Config.  
    
    
  
* Enable AWS Security Hub. Allow two hours for it to setup before use.  
    
    
    
  

Windows Server 2019 Setup:

* In the free tier of EC2, launch a new instance of Windows Server 2019.  
  
  + In order to ensure you stay within the free tier, select t2.micro under instance type.  
      
    
* Establish remote connectivity to your new instance via RDP, this will require you give your instance an Elastic IP (public IP address)  
  
* Download the system-hardening.ps1 script to your Windows Server 2019 instance.
  + Use a chatbot to breakdown for you what the system-hardening.ps1 script is doing.  
    **●** **doing.**
  + The system-hardening.ps1 script has a misleading name because it actually weakens a system's security posture. Here's a breakdown of what it likely does:
  + **Disabling Security Features:**
  + ● **Windows Defender:** The script disables real-time protection, potentially creating a vulnerability for malware to infect the system.
  + ● **Security Policies (Commented Out):** A commented section suggests it might modify security policies, which can significantly impact security if not done carefully.
  + ● **User Account Control (Commented Out):** Another commented section seems to disable User Account Control (UAC) prompts, allowing unauthorized programs to run without warnings.
  + **Creating a Vulnerable User Account:**
  + ● It adds a new local user named "zuko" with administrator privileges. This can be risky in a production environment as administrator accounts have full control over the system.
  + **Introducing Potential Threats:**
  + ● **Downloading Unknown File:** The script might download a compressed file from the internet. This file could potentially contain malware, and the script extracts it to a critical system directory, further increasing risk.
  + **Restarting the System:**
  + ● After making these changes, the script restarts the system, applying the configurations that weaken security.
  + **Important Considerations:**
  + ● **Training Purposes Only:** This script is likely intended for a controlled training environment to demonstrate security vulnerabilities. Running it on a real system you want to protect is highly discouraged.
  + ● **Ethical Hacking Practices:** Modifying security measures on someone else's system without permission is unethical hacking. Always get consent before running such scripts.
  + ● **Alternative Learning Methods:** Consider using virtual machines for security training to avoid compromising a real system.
  + In conclusion, the system-hardening.ps1 script is a deceptive tool designed to create a vulnerable environment for security training purposes. It's important to understand its functionality and only use it in a controlled setting.
* Disable Windows Defender and set the execution policy to allow the script to run using the following commands:
  + Set-MpPreference -DisableRealtimeMonitoring $True
  + Set-ExecutionPolicy Bypass -Force
* Run PowerShell as administrator and navigate to the directory containing the system-hardening.ps1 script
* Run the script with the .\system-hardening.ps1 command
* The script will run and restart your instance (you will lose connection)
* The script will create an admin account with the following credentials:
  + Username: zuko
  + Password: R0cks!  
    
* RDP to your instance using these credentials and complete the lab  
  

### **Part 2: Manual EC2 instance hardening with CIS**

Note: There are CIS-compliant AMIs that are prehardened, but for this part of the lab use regular AMIs.

* Select the following six benchmarks from your AMI’s benchmark document in the [CIS Microsoft Windows Server 2019 Benchmark v2.0.0](https://downloads.cisecurity.org/#/) and reconfigure your Windows Server instance, using the GUI via RDP, to achieve the standard indicated.  
    
  + 1.1.5 (L1)  
    
  + 1.1.6 (L1)
  + 1.2.1 (L1)
  + 1.2.2 (L1)
  + 18.4.3 (L1)
  + 18.4.4 (L1)
* Include a screenshot of each configuration change in your instance session, including a full description of the requirement as indicated in the CIS document.

### **Part 3: Automating the Configuration**

In some scenarios, you may need to harden an existing machine. Automating the process will save you a great deal of time in the long run.

* Reverse your configuration changes to:
  + 1.1.5 (L1)
  + 18.4.3 (L1)
* Write a PowerShell script that automates the configuration of the required settings:
  + 1.1.5 (L1)
  + 18.4.3 (L1)
* Test and validate it works as desired.
* Copy it to your public GitHub repo as a new entry. Link to it in your submission.

### **Part 4: Pre-hardened EC2 instance deployment**

Now let’s compare this to a pre-hardened AMI.

* Deploy a pre-hardened AMI in AWS EC2. Ideally selected the hardened version of the OS you just worked on.
* Access the CIS standards and verify the six benchmarks are achieved on the hardened instance:
  + 1.1.5 (L1)
  + 1.1.6 (L1)
  + 1.2.1 (L1)
  + 1.2.2 (L1)
  + 18.4.3 (L1)
  + 18.4.4 (L1)
* Take a screenshot verifying each benchmark in your pre-hardened instance.
* Does the hardened instance achieve the security standard on deployment?

### **Part 5: Scanning for CIS Benchmarks**

Next, navigate to Security Hub and post the CIS dashboard in your submission doc.

* Both AMIs should be running.
* Wait for an automated CIS scan.
* Paste your findings (security score visual and output table) into your submission doc.
* Assess your findings. Make at least one change that improves the result of your CIS scan.
* Report on your efforts (assessment, action, result), and recommend an SOP for company usage of EC2.

Note: There is no way to force Security Hub to run a scan. Security Hub rus the standard checks within 2 hours after enabled and checks again automatically within 12 hours from the latest scan.

### **Part 6: Wrap-up**

Stop AWS resources deployed today in order to avoid charges. We will, however, be using AWS Security Hub (and other AWS tools) again in this course.