# **Purpose**

The purpose of this deployment was to create my own Jenkins servicer and then manually deploy to AWS EB.

# **Steps**

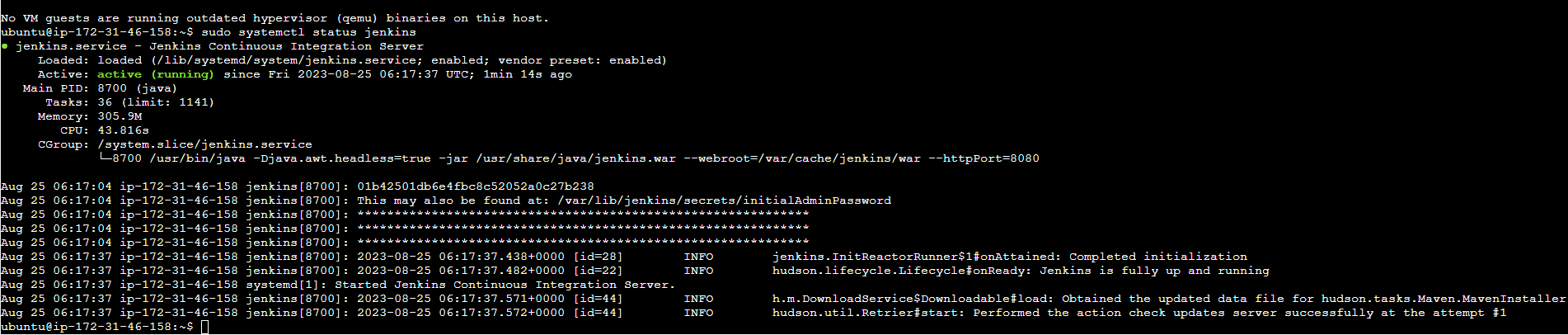
1. Before I started the instructions below, I first started to create my diagram in Draw.io (this will beupdated as I goalong with the project).
2. From there I created my own GitHub repository which included a README.md file for documentation.
3. Once my GitHub repo was created, I downloaded the zip files and then uploaded them to my repo:
   * Download zip files
   * Unzip into new folder in File Explorer
   * Go into the folder and select all files
   * Copy and paste, or drag files into GH repo
   * Commit changes (save)
4. Then I created an EC2 terminal
5. Establish connection between EC2 and local terminal
6. Create Jenkins server
7. Install "python3.10-venv"
8. Download Jenkins server:
   * use public IP address from EC2 instance and put into web browser along with :8080 (example: 54.242.39.43:8080)
   * **NOTE:** If I close my EC2 instance and reopen again, I will have another public IP address. Use the this new IP address and :8080 to log in.
9. Install the Jenkins plugin "Pipeline Utility Steps"
10. Create and run a Jenkins build for the application
11. Observe pipeline stages via the console output and document what occurred

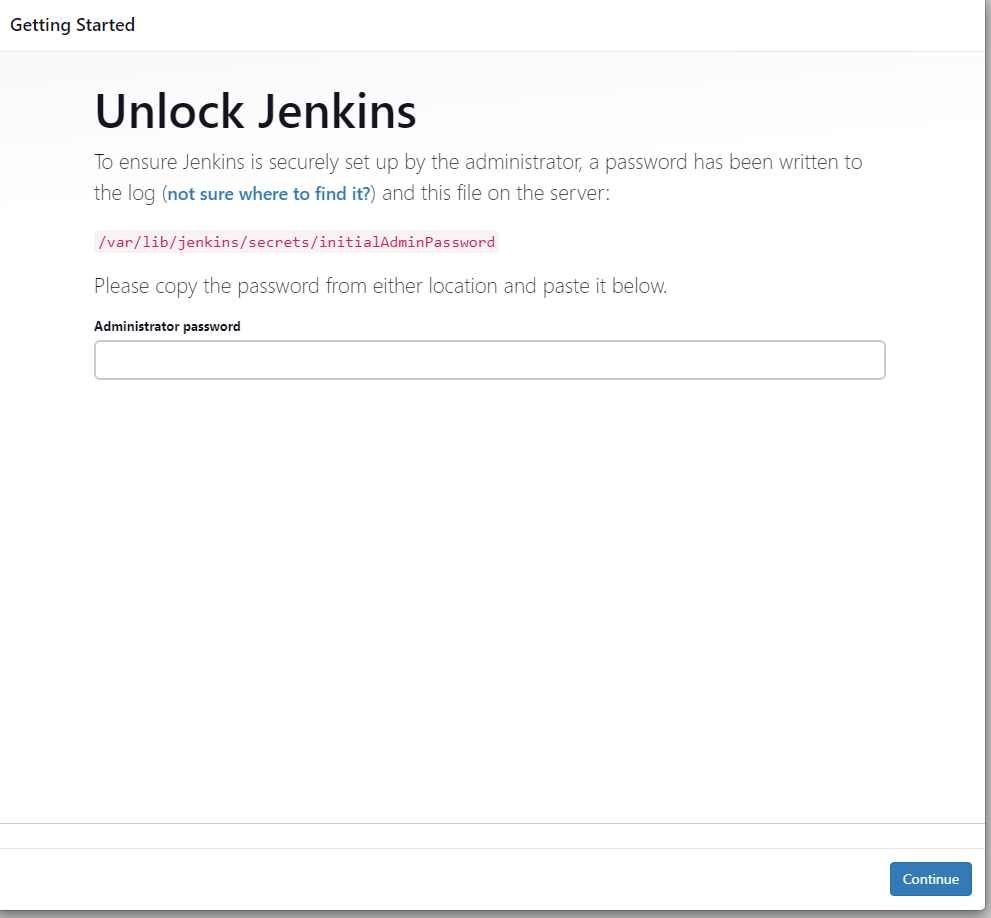


1. Create an environment and url using AWS EB: <http://url-shortener-env.eba-vpn8m7zw.us-east-1.elasticbeanstalk.com/deployment2>

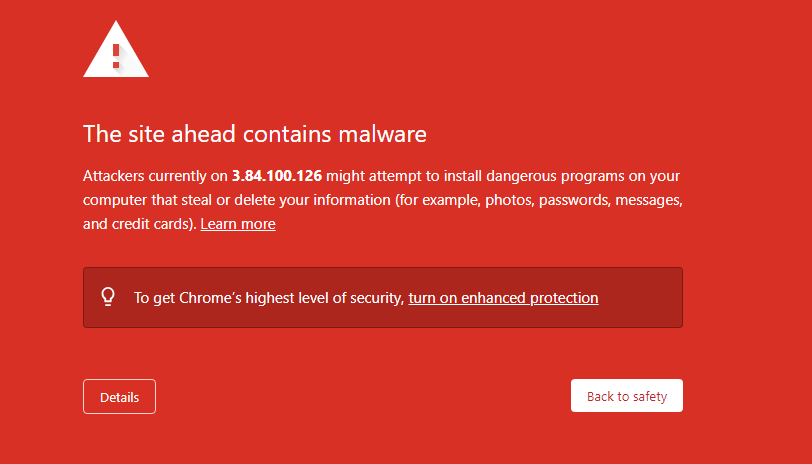
# **Issues**

1. Connecting my EC2 instance using SSH. I kept getting permission denied, and therefore could not create a Jenkins server to my machine. How did I fix this? To remember this going forward ALWAYS make sure I have my public and private keys in my local terminal and in my ec2 instance.
2. Creating my Jenkins server. Once I fixed my ssh connection, I was able to install Jenkins successfully.





1. WAS UNABLE TO ACCESS JENKINS THE NEXT DAY AFTER GETTING THIS BIG, SCARY RED SCREEN BEFORE REALIZING THAT ALL I HAD TO DO WAS CLICK DETAILS > Visit this unsafe site



# **Optimization** (How would make this deployment more efficient)

# I would find a way to make the process a bit faster when packaging the output files. For example, if it takes more than, let’s say 5 minutes, have a pop up to ask if I want to abort or continue, instead of having to manually do this. If I hadn’t known any better, I would have been waiting forever and thinking that something was wrong on my end.



# **System Diagram**

To view the diagram of the system design pipeline, click HERE

