October 2026

Build, Lure, Learn – Workshop Instructions

1. AWS Setup

- Either create an account or login:https://console.aws.amazon.com/
- Then add a valid payment method

2. Deploy an EC2 Instance

Navigate to **EC2**

1. Make note of your region (ex: US N. Ohio | us-east-2)
2. Click on 'Launch instance'
    a. Enter Name (ex: My Server)
    b. Under 'Application and OS Images' -> Quick Start –> Select Ubuntu (pick latest version)
    c. 'Instance type' -> 't3a.xlarge' (confirm 16 gb of memory)
    d. 'Key pair (login)' -> Create a new key pair -> provide a name (ex: keys)
        i. Leave 'key pair' as is
        ii. Use .pem for key file format
            1. expect to use .pem unless you know you need to use PuTTY
        iii. Click 'Create key pair', download and save them (take note of where you saved them)
    e. For 'Network settings', under 'Firewall' -> confirm following selection
        i. 'Create security group' is selected
        ii. 'Allow SSH traffic from' -> **My IP**
    f. For 'Configure storage' -> change to capacity 256 and leave gp3 as is
3. Click Launch Instance

3. Create Role for Budget Alert

While we wait for instance to initialize, let's create the budget and set an alert. As part of the alert, we will want to add an action so that once the threshold of our budget has passed, it will automatically stop the instance we spun up. To do such a thing, we will need to have a role associated to that action with the privileges to stop an instance.

1. Navigate to 'IAM' -> 'Roles' then click 'Create role'
    a. For 'Select trusted entity type' -> AWS Service
    b. For 'Use case' -> 'Service or use case' -> select 'Budgets'
    c. Click next
    d. For 'Add permissions', select the 'AWSBudgetsActions_RolePolicyForResourceAdministrationWithSSM' policy
    e. Click next

f. For ‘Name, review, and create’, add a name for this role (ex: BudgetStopEC2Role), you can change the description and leave the rest as is
g. Click ‘Create Role’

4. Create Budget & Alert

Now that we have the required role, let’s go create our budget and alert

1. Navigate to ‘Billing and Cost Management’ -> ‘Budgets’ -> ‘Create a Budget’
    a. Select ‘Customize (advanced)’
        1. For ‘Choose budget Type’, select ‘Cost budget’ then click next
    ii. Under ‘Set your budget’
        1. Add budget name
        2. Set ‘Period’ (recommend ‘monthly’)
        3. Leave ‘Budget Renewal type’, ‘Start month’, and ‘Budgeting Method’ as is (‘Recurring budget, Oct 2025, Fixed)
        4. Enter your budgeted amount -> $10 *should be enough for this weekend* (Expect ~$80 for around a month)
        5. Leave ‘Budget Scope’ as is (All AWS services, Unblended costs)
        6. No need to add tags, click next
    iii. Under ‘Configure alerts’
        1. Add an alert threshold,
            a. Set Threshold amount to %80 (Why? Even after we stop the instance, the storage will still be charged so we will leave a bit of a buffer)
            b. Leave rest as is, add your email for notifications, then click next
        2. Attach action
            a. On the alert threshold we just added -> click on ‘Add action’
            b. ‘Select IAM role’ -> select role we just created
            c. ‘Which Action type should be applied when the budget threshold has been exceeded?’ -> select the option ‘Automate instances to stop for EC2 or RDS’
            d. For ‘Stop EC2/RDS instances’ -> select ‘Stop EC2’
            e. For ‘Select Region’, ‘choose EC2/RDS instance’ -> click on the region you deployed your instance in then select your instance
            f. For the question ‘Do you want to automatically run this action when….?’, select ‘Yes’
            g. Leave rest as is and click next
        3. Review and then click Create budget
            a. Please remember this alert will help stop the charges occurring but it won’t completely remove them. You can

review the 'Clean Up' section to learn how to completely remove everything spun up to stop charges occurring.

5. Deploy T-Pot

Link to T-Pot repo: https://github.com/telekom-security/tpotce
After creating our budget and alert, we can go back to our EC2 instance and install T-Pot

- Navigate to 'EC2' -> 'Instances' -> click on our new instance -> click 'Connect' -> 'SSH Client'
    a. Ensure your keys are as said (otherwise run the chmod command)
    b. Copy the command for SSH and open command prompt (or terminal)
    c. Paste and enter. You will get back a response along the lines of "…are you sure you want to continue connecting?.." -> enter 'yes'
        i. Connection may close, just run SSH command again
    d. Run the following commands to update the packages and install git
        i. sudo apt-get update (or sudo apt update -y)
        ii. sudo apt-get install git (or sudo apt install git -y) (git may be already installed)
    e. Then clone the T-Pot repository -> git clone https://github.com/telekom-security/tpotce
    f. After it has been cloned, cd tpotce/
    g. Run installer script by doing entering ./install.sh
    h. Once you get the question 'Install?', type y
    i. For 'Choose your T-Pot Type' -> type h
    j. For 'Enter your web user name:' -> enter a username you want (remember you will need this to later log into the T-Pot web UI)
    k. Enter a password you want
    l. After the installation is done, note that moving forward you'll have to reconnect via **tcp/64295**
    m. Enter 'sudo reboot' to reboot instance, connection will close

6. Update EC2 Instance Security Group

Because the installation changed our SSH port to 64295, we need to change our security group rules to reconnect. We will also add additional rules.

1. Navigate to your instance -> 'Security' -> click on the security group -> click on 'Edit inbound rules'
2. Update inbound rules as follows
    a. We can delete the rule already there
    b. Add a rule for our T-Pot SSH –> Type (Custom TCP), Port Range (64295), Source (My IP), Description (SSH for T-Pot)

c. Add a rule for our T-Pot Web UI–> Type (Custom TCP), Port Range (6429**7**), Source (My IP), Description (Web Portal for T-Pot)
d. Add a rule for incoming IPv4 TCP traffic –> Type (Custom TCP), Port Range (1 - 64000), Source (Anywhere-IPv4 (0.0.0.0/0)), Description (Attack Surface (IPv4 UDP))
e. Add a rule for incoming IPv4 UDP traffic –> Type (Custom UDP), Port Range (1 - 64000), Source (Anywhere-IPv4 (0.0.0.0/0)), Description (Attack Surface (IPv4 UDP))

7. Confirm T-Pot is Running

Now that we have our security group configured, lets go back and reconnect through SSH to confirm our T-Pot is up and running

1. Go to your command prompt/terminal and enter the command we used to first connect via SSH but add the correct port
    a. ssh -i your-key.pem ubuntu@ec2-xx-xxx-xxx-xxx.us-east-2.compute.amazonaws.com **-p 64295**
    b. Type dps and check everything is up and running

8. Access T-Pot Web UI

Now let's try to access our T-Pot via its web UI

1. Go back to your EC2 instance
2. Go down to 'details' -> under 'Public IPv4 address' click on 'open address'
3. When new tab is opened, add the port at the end of the url (**:64297)** and then login with your credentials
4. Now you should be logged in and you can start exploring the attacker data and provided tools

9. Explore The Data

You may need to let the T-Pot sit for a bit to collect data, but once it starts populating, you may begin exploring it. Some examples of what you can do

- Review available Kibana dashboard
- Review available attack map
- Take an attacker IP and check for more information in Spiderfoot
- Review usernames/passwords attackers are trying to use

October 2026

10. Clean Up

Once you finish with this project, you can proceed with the project clean up. There are three main items to review:

1. The EC2 instance - this costs money
2. The storage (EBS volumes) attached to the EC2 instance - this costs money and may continue to incur storage charges even after the EC2 instance is terminated
3. The budget with action - depending on how many action-enabled budgets you have, there may be additional charges

11. Terminate the EC2 Instance

1. Navigate to 'EC2' -> Select Instance -> Under 'Instance state' -> select 'Terminate (delete instance)'
2. Confirm instance has been terminated (shown under 'Instance State')

12. Verify and Delete Unneeded Storage

Do **not** assume that terminating theEC2 instance automatically deletes its attached storage!

1. Navigate to 'EC2' -> 'Elastic Block Store' -> 'Volumes'
2. Select the associated Volume -> 'Actions' -> 'Delete Volume'
3. Confirm the volume is deleted/not there anymore

13. Review the Budget with an Action

Depending on how many budgets with actions you have, it may keep incurring charges so to be on the safe side you can delete the budget by:

1. Navigate to 'Cost and Billing Management' -> 'Budgets and Planning' -> 'Budgets'
2. Select the budget with an action enabled -> 'Actions' -> 'Delete'
3. Confirm the budget was deleted

To be sure of no additional charges occurring, you may want to keep the simple 'My Zero-Spend Budget' (template is available)

Monitor for Additional Charges

Even after cleanup, continue monitoring your AWS account to ensure that no unexpected charges occur.

Remember:

- Check current or previous charges in the Billing and Cost Management
- Check all AWS you used for any remaining resources you may have missed in the cleanup

- Billing information may take time to update. There may be charges that appear after cleanup that were related to resources used earlier in the billing period.

Next Steps (Links)

CTI

- A Cyber Threat Intelligence Self-Study Plan: Part 1 by Katie Nickels (Medium)
- Cyber Threat Intelligence Study Plan by Andy Piazza (Medium)

Honeypots & Cyber Deception

- MITRE Engage Framework
- Awesome Honeypots
- Cyberdeception Atlas
- (Book) The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage by Clifford Stoll

AWS/Cloud

- AWS Documentation
- AWS Skills Builder