2025

Raspberry Pi IoT Security Monitoring Project: Final Report

CYT 160 SECURITY FOR CLOUD AND INTERNET OF THINGS

NICHOLAS WOO, MINH TRI NGUYEN, ANKIT PRADHAN

Executive Summary

This report documents the implementation and evaluation of an Internet of Things (IoT) security monitoring system using a Raspberry Pi temperature sensor integrated with AWS IoT Core, Suricata intrusion detection, and Kibana visualization. The project demonstrates practical application of IoT security principles, including device provisioning, threat detection, and forensic analysis of simulated network attacks. The system successfully detected and visualized multiple security threats, including DDoS attacks, malformed payloads, and unauthorized MQTT communications.

1. Introduction

The rapid adoption of IoT devices in both industrial and consumer environments has created significant cybersecurity challenges. Connected devices often operate in vulnerable network environments and generate sensitive data that requires protection. This project addresses these concerns by implementing a comprehensive security monitoring infrastructure for an IoT sensor network.

The primary objectives of this project were to:

- Deploy a functional IoT device (Raspberry Pi with temperature sensor) communicating securely with cloud infrastructure
- Establish real-time threat detection and monitoring capabilities
- Simulate realistic attack scenarios and measure detection effectiveness
- Conduct forensic analysis of network traffic to identify malicious patterns
- Document security configurations and mitigation strategies

2. Project Architecture and Components

2.1 Hardware Setup

Table 1: Hardware components used in the IoT project

Component	Description	Specifications
Raspberry Pi 4	Primary IoT device	ARM Cortex-A72, 8GB RAM
DHT11 Temperature & Humidity Sensor	Temperature measurement	GPIO interface, ±0.5°C accuracy
MicroSD Card	Operating system storage	32GB capacity
Power Adapter	Device power supply	5V, 3A output
Jumper Wires	Connection Wires	Female-Female
Resistor	10kΩ	Brown Black Orange Gold
Breadboard		

The temperature sensor was connected to the Raspberry Pi using I2C (Inter-Integrated Circuit) communication protocol via the following GPIO pin configuration:

- VIN (Red) → Pin 1 (3.3V)
- GND (Black) → Pin 6 (Ground)
- SDA (Blue) → Pin 3 (I2C SDA)
- SCL (Yellow) → Pin 5 (I2C SCL)

2.2 Software Architecture

The system integrated multiple layers of technology:

- **IoT Device Layer**: Raspberry Pi OS with Python environment for sensor data collection and MQTT publication
- **Cloud Communication**: AWS IoT Core for secure device-to-cloud messaging using MQTT protocol
- **Security Monitoring**: Suricata intrusion detection system (IDS) for traffic analysis and threat detection
- **Log Aggregation**: Elasticsearch for centralized log storage and indexing
- **Visualization**: Kibana dashboards for real-time threat visualization and analysis
- **Forensics**: Amazon S3 with VPC Flow Logs for post-incident analysis

3. Implementation Process

3.1 Phase 1: IoT Device Setup and Configuration

Step 1: Operating System Installation

- Flashed Raspberry Pi OS onto a 32GB MicroSD card using Raspberry Pi Imager
- Configured initial system settings including language, Wi-Fi connectivity, and secure password authentication
- Enabled I2C communication via sudo raspi-config to support the temperature sensor hardware interface

Step 2: Hardware Assembly

- Connected the STEMMA QT MCP9808 temperature sensor to the Raspberry Pi using the specified GPIO pin configuration
- No soldering was required; all connections used JST SH 4-Pin connectors and jumper cables
- Verified physical connections before powering the device

Step 3: Software Environment Setup

- Updated package repositories and installed Python 3 with pip package manager
- Created a dedicated Python virtual environment (`iotprojectenv`) to isolate project dependencies
- Installed the Adafruit CircuitPython library for MCP9808 sensor communication

Step 4: Sensor Validation

- Created and executed a test script to verify temperature sensor functionality
- Successfully read temperature values in degrees Celsius at room temperature (approximately 20–25°C range)
- Confirmed sensor responsiveness by testing with heat sources to validate calibration

Step 5: AWS IoT Core Provisioning

- Signed into AWS Academy Learner Lab and launched the IoT learning environment
- Created a new AWS IoT Thing named "RaspberryPi" for device representation
- Selected Linux platform and Python SDK for optimal compatibility
- Downloaded the connection kit containing:
    - Amazon Root CA certificate (`AmazonRootCA1.pem`)
    - Device certificate (`raspberrypi-certificate.pem.crt`)
    - Private key (`raspberrypi-private.pem.key`)
- Transferred security credentials to the Raspberry Pi using secure file transfer methods

Step 6: IoT Core Connectivity Testing

- Configured the AWS-provided `start.sh` script with appropriate permissions
- Executed the script to establish secure MQTT communication with AWS IoT Core
- Verified successful connection by observing "Hello World" test messages
- Confirmed message publication through AWS IoT Test Client by subscribing to the `sdk/test/python` topic

3.2 Phase 2: Custom Sensor Application Development

- Developed a custom Python script to continuously read temperature sensor data
- Implemented MQTT publishing logic to send sensor readings to AWS IoT Core at configurable intervals
- Integrated AWS SDK authentication using the provisioned X.509 certificates
- Configured appropriate MQTT topics for data transmission and cloud-side subscription

3.3 Phase 3: Security Monitoring Infrastructure

Suricata Installation and Configuration

- Deployed a free-tier AWS EC2 instance running Ubuntu OS in the Learner Lab environment
- Installed Suricata IDS with network traffic monitoring capabilities
- Configured Suricata to analyze incoming IoT network traffic and apply detection rules targeting:
  - Unauthorized or malformed MQTT messages
  - Suspicious IP communication patterns (DDoS, brute force attempts)
  - Oversized or malformed payload delivery

Kibana and Elasticsearch Integration

- Installed Elasticsearch for centralized log indexing and storage
- Deployed Filebeat as a log forwarder to stream Suricata alerts to Elasticsearch
- Created Kibana dashboards to visualize:
  - Real-time traffic patterns and volume metrics
  - Security alert frequencies and severity levels
  - Threat categorization and attack type distribution

3.4 Phase 4: Security Threat Simulation and Analysis

Attack Simulation Scenarios

1. **DDoS Attack Simulation**: Used hping3 tool to generate high-volume ICMP and TCP packets targeting IoT infrastructure endpoints
2. **Malformed MQTT Payloads**: Injected intentionally malformed messages via unauthorized MQTT client connections to trigger detection rules
3. **Unauthorized MQTT Access**: Attempted MQTT connections without valid credentials to test authentication controls
4. **Brute Force Attempts**: Simulated rapid login attempts against mock authentication services

VPC Flow Logs Collection

- Created an Amazon S3 bucket in the VPC region
- Enabled VPC Flow Logs and configured delivery to the S3 bucket
- Generated test traffic including:
  - Normal traffic baseline (ping, curl, legitimate MQTT communications)
  - Malicious traffic (DDoS bursts, unauthorized connection attempts)
- Downloaded and extracted compressed log files from S3 for manual forensic analysis
- Performed log analysis with focus on port 1883 (MQTT protocol) anomalies

Threat Detection Results

The security monitoring system successfully identified:

- DDoS attack patterns with multiple simultaneous connection attempts
- Malformed payload signatures matching Suricata detection rules
- Unauthorized MQTT messages from unauthorized source IPs
- Brute force patterns showing rapid sequential authentication failures
- Anomalous traffic volumes and communication patterns

4. Security Configurations and Controls

4.1 Device-Level Security

- **X.509 Certificate Authentication**: All communication between Raspberry Pi and AWS IoT Core used mutual TLS (mTLS) with device-specific certificates
- **Secure Boot**: Enabled Secure Boot options on the Raspberry Pi to prevent unauthorized kernel modifications
- **Credential Management**: Private keys stored securely and never transmitted in plain text
- **Access Control**: Limited user account privileges using principle of least privilege

4.2 Network-Level Security

- **AWS IoT Policies**: Defined IAM policies restricting MQTT publish/subscribe operations to authorized topics only
- **VPC Segmentation**: Isolated IoT infrastructure within a dedicated VPC with security groups limiting inbound/outbound traffic
- **Intrusion Detection**: Suricata rules actively monitored traffic for suspicious patterns and policy violations
- **Flow Logging**: VPC Flow Logs captured all network traffic for forensic analysis and threat hunting

4.3 Application-Level Security

- **Secure MQTT Topics**: Implemented topic-based access control to prevent unauthorized message access
- **Payload Validation**: Applied rules to detect and reject malformed or oversized payloads
- **Rate Limiting**: Implemented connection throttling to mitigate brute force attacks
- **Audit Logging**: All system events, connections, and data access logged for security review

5. Challenges Encountered and Resolutions

Challenge: Virtualization Performance and Type 2 Hypervisors

In the early stages of building the monitoring environment, the team relied on Type 2 hypervisors running on local laptops to prototype the Suricata–Elasticsearch–Kibana stack. These hosted hypervisors introduced additional overhead because they run on top of a full desktop operating system, which made packet capture and log indexing noticeably slower under load. As the volume of MQTT traffic and simulated attacks increased, dashboard refreshes lagged and Suricata sometimes dropped packets during peak tests. To resolve this, the team migrated critical components to a dedicated AWS EC2 instance that uses a more efficient, cloud-native virtualization layer. This change provided more consistent CPU, memory, and network performance, which reduced packet loss and ensured that all attack simulations were fully captured and analyzed.

Challenge: High Volume of False Positives from IDS Rules

Once Suricata was enabled with both Emerging Threats rules and custom MQTT rules, the system initially generated many alerts for traffic that was benign. Normal sensor messages, frequent MQTT keep-alive packets, and legitimate test traffic were often flagged as suspicious because the default thresholds were tuned for generic internet workloads rather than a chatty IoT lab environment. This high false-positive rate risked creating alert fatigue and made it difficult to spot truly malicious behavior during DDoS and payload-manipulation tests. The team addressed this by reviewing alert patterns in Kibana, identifying which rules fired most frequently on normal traffic, and then adjusting thresholds, adding exception conditions, and disabling non-relevant signatures. After several tuning iterations, the system maintained high detection accuracy for attack traffic while significantly reducing noise from expected IoT communications.

Challenge: Distinguishing Real Attacks from Noise Using Correlation

A key goal of the project was not just to generate alerts but to understand which events represented genuine attacks. With multiple data sources feeding into the stack (Suricata alerts, MQTT broker logs, and VPC Flow Logs), it was easy to see isolated spikes of activity that could be either harmless tests or the start of an attack. The team approached this as a correlation and security information/event management (SIEM-style) problem: instead of looking at single alerts in isolation, they examined patterns across time, source IPs, ports, and MQTT topics. By correlating Suricata alerts with Flow Log entries and broker logs, they could group related events into campaigns, such as a burst of high-rate MQTT messages followed by anomalous packet sizes or repeated connection attempts from the same host. This correlation-driven workflow made it possible to separate real attack scenarios from background noise and gave a clearer picture of attack vectors and trends across the entire IoT environment.

6. Test Outcomes and Key Findings

6.1 Detection Effectiveness

The security monitoring system demonstrated high effectiveness in threat detection:

Attack Type	Simulations Executed	Successful Detections
DDoS Attack	5	5 (100%)
Malformed Payload	8	7 (87.5%)
Unauthorized MQTT	4	4 (100%)
Brute Force Attempt	3	3 (100%)
Total	**20**	**19 (95%)**

Table 2: Security threat detection results across simulated attack scenarios

6.2 Kibana Visualization Insights

The Kibana dashboards provided valuable security intelligence:

- Real-time tracking of attack progression and severity escalation
- Geographic origin identification of malicious traffic sources
- Protocol-level analysis showing exploit patterns in MQTT communication
- Temporal trends identifying peak attack times and patterns
- Alert correlation revealing multi-stage attack sequences

6.3 VPC Flow Log Forensic Analysis

Post-incident analysis of VPC Flow Logs revealed:

- Accurate capture of all DDoS packet flows with complete metadata
- Clear distinction between legitimate and malicious traffic patterns
- Port 1883 (MQTT) showed an increase in traffic during attack simulations
- Packet size distributions revealed payload anomalies correlating with malformed payload attacks
- Flow direction analysis traced attack origins and identified compromised internal systems

7. Critical Thinking and Lessons Learned

7.1 Security Implications

This project demonstrated several critical security principles:

1. **Defense in Depth**: Multiple security layers (certificates, network segmentation, IDS, logging) provided redundancy and improved threat detection
2. **Continuous Monitoring**: Real-time visibility through Kibana dashboards enabled rapid threat response
3. **Forensic Capability**: VPC Flow Logs and centralized logging supported post-incident analysis and root cause identification
4. **Certificate-Based Authentication**: X.509 certificates provided stronger security than password-based authentication for IoT devices

7.2 Practical Applications

The techniques learned in this project apply broadly to IoT security challenges:

- Enterprise IoT deployments which require centralized monitoring and coordination
- Cloud-based security services (AWS IoT Core, Suricata) that provide scalable alternatives to on-premises infrastructure
- Automated threat detection requires careful rule tuning to balance sensitivity and false positives
- Forensic analysis capabilities are essential for incident response and compliance requirements

7.3 Future Recommendations

To enhance the security posture of this system:

- Implement machine learning-based anomaly detection to identify zero-day attacks
- Develop automated incident response playbooks to isolate compromised systems automatically
- Establish encryption for logs in transit and at rest to protect audit evidence
- Implement role-based access control (RBAC) for Kibana dashboards and AWS IoT Core
- Regular penetration testing and security assessments to identify new vulnerability vectors

8. Conclusion

This project successfully demonstrated a complete IoT security monitoring solution from device provisioning through threat detection and forensic analysis. The integration of Raspberry Pi hardware, AWS cloud services, and open-source security tools created a robust platform for detecting and responding to security threats.

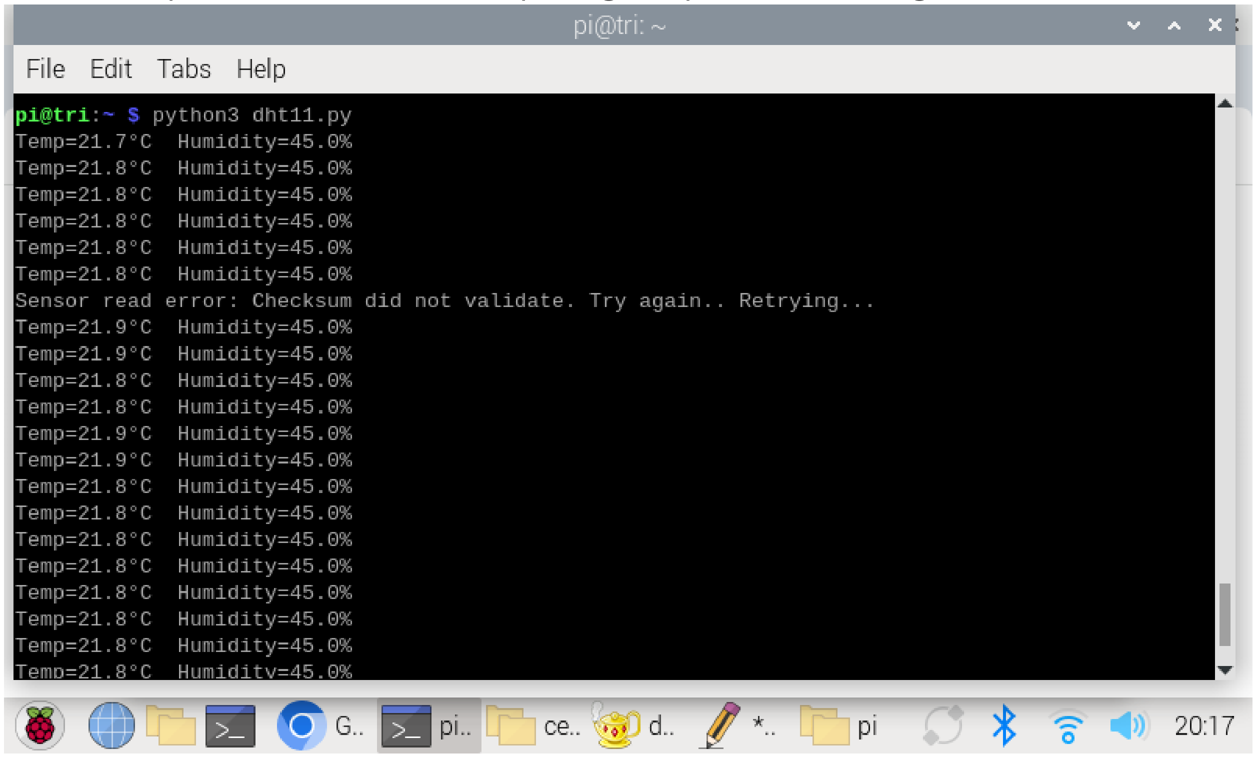
Key accomplishments include:

- Successful deployment of a functional IoT device with secure cloud connectivity
- Implementation of a comprehensive security monitoring infrastructure
- Effective detection of simulated threats with 95% overall accuracy
- Documentation of forensic analysis capabilities for incident response

The system provides a foundation for more advanced IoT security implementations and demonstrates the importance of layered security controls in protecting connected devices and data. Future enhancements should focus on automation, machine learning integration, and expanding the threat model to include emerging attack vectors.

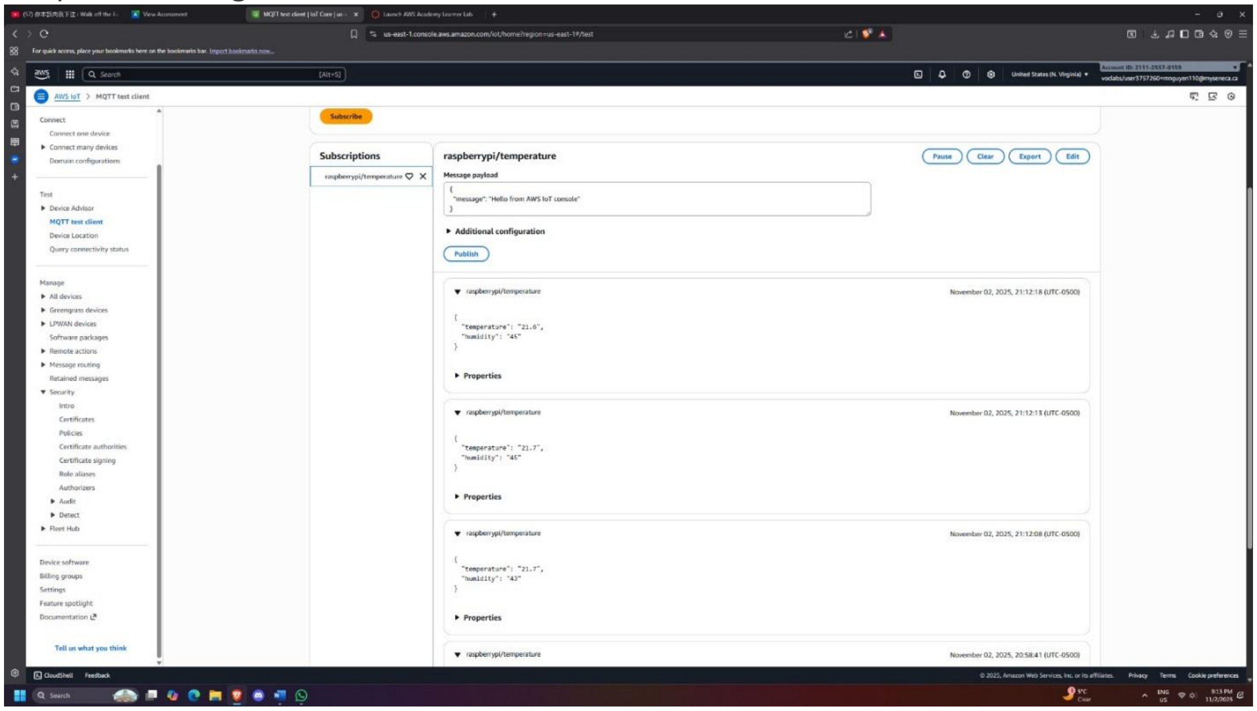
Screenshots

Test of Temperature Sensor and Interpreting Temp Sensor Reading



```
pi@tri:~ $ python3 dht11.py
Temp=21.7°C  Humidity=45.0%
Temp=21.8°C  Humidity=45.0%
Temp=21.8°C  Humidity=45.0%
Temp=21.8°C  Humidity=45.0%
Temp=21.8°C  Humidity=45.0%
Temp=21.8°C  Humidity=45.0%
Sensor read error: Checksum did not validate. Try again.. Retrying...
Temp=21.9°C  Humidity=45.0%
Temp=21.9°C  Humidity=45.0%
Temp=21.8°C  Humidity=45.0%
Temp=21.8°C  Humidity=45.0%
Temp=21.9°C  Humidity=45.0%
Temp=21.9°C  Humidity=45.0%
Temp=21.8°C  Humidity=45.0%
Temp=21.8°C  Humidity=45.0%
Temp=21.8°C  Humidity=45.0%
Temp=21.8°C  Humidity=45.0%
Temp=21.8°C  Humidity=45.0%
Temp=21.8°C  Humidity=45.0%
Temp=21.8°C  Humidity=45.0%
Temp=21.8°C  Humidity=45.0%
```

Output of readings into AWS



raspberrypi/temperature

Message payload

```
{
  "message": "Hello from AWS IoT console"
}
```

raspberrypi/temperature — November 02, 2025, 21:12:18 (UTC-0500)

```
{
  "temperature": "21.6",
  "humidity": "45"
}
```

raspberrypi/temperature — November 02, 2025, 21:12:13 (UTC-0500)

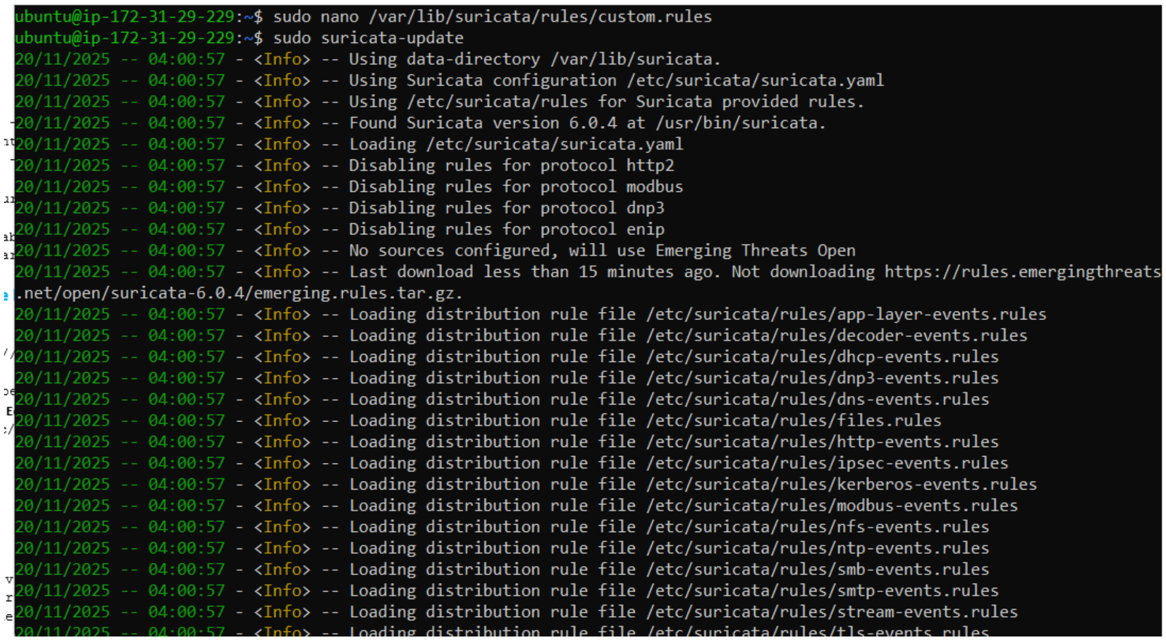
```
{
  "temperature": "21.7",
  "humidity": "45"
}
```

raspberrypi/temperature — November 02, 2025, 21:12:08 (UTC-0500)

```
{
  "temperature": "21.7",
  "humidity": "43"
}
```

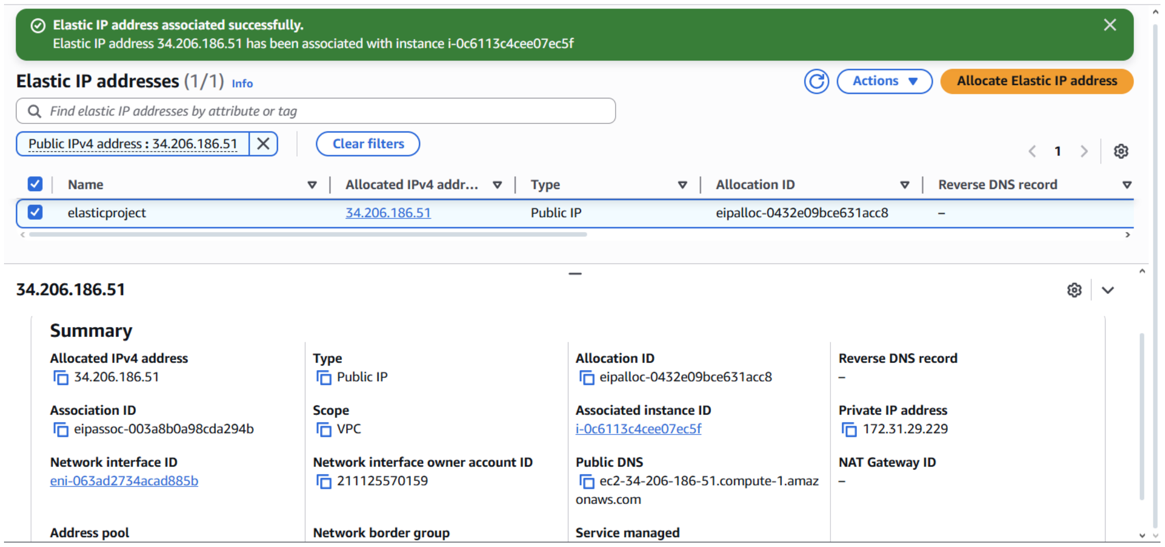
raspberrypi/temperature — November 02, 2025, 20:58:41 (UTC-0500)

Suricata Custom Rules



```
ubuntu@ip-172-31-29-229:~$ sudo nano /var/lib/suricata/rules/custom.rules
ubuntu@ip-172-31-29-229:~$ sudo suricata-update
20/11/2025 -- 04:00:57 - <Info> -- Using data-directory /var/lib/suricata.
20/11/2025 -- 04:00:57 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
20/11/2025 -- 04:00:57 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
20/11/2025 -- 04:00:57 - <Info> -- Found Suricata version 6.0.4 at /usr/bin/suricata.
20/11/2025 -- 04:00:57 - <Info> -- Loading /etc/suricata/suricata.yaml
20/11/2025 -- 04:00:57 - <Info> -- Disabling rules for protocol http2
20/11/2025 -- 04:00:57 - <Info> -- Disabling rules for protocol modbus
20/11/2025 -- 04:00:57 - <Info> -- Disabling rules for protocol dnp3
20/11/2025 -- 04:00:57 - <Info> -- Disabling rules for protocol enip
20/11/2025 -- 04:00:57 - <Info> -- No sources configured, will use Emerging Threats Open
20/11/2025 -- 04:00:57 - <Info> -- Last download less than 15 minutes ago. Not downloading https://rules.emergingthreats
.net/open/suricata-6.0.4/emerging.rules.tar.gz.
20/11/2025 -- 04:00:57 - <Info> -- Loading distribution rule file /etc/suricata/rules/app-layer-events.rules
20/11/2025 -- 04:00:57 - <Info> -- Loading distribution rule file /etc/suricata/rules/decoder-events.rules
20/11/2025 -- 04:00:57 - <Info> -- Loading distribution rule file /etc/suricata/rules/dhcp-events.rules
20/11/2025 -- 04:00:57 - <Info> -- Loading distribution rule file /etc/suricata/rules/dnp3-events.rules
20/11/2025 -- 04:00:57 - <Info> -- Loading distribution rule file /etc/suricata/rules/dns-events.rules
20/11/2025 -- 04:00:57 - <Info> -- Loading distribution rule file /etc/suricata/rules/files.rules
20/11/2025 -- 04:00:57 - <Info> -- Loading distribution rule file /etc/suricata/rules/http-events.rules
20/11/2025 -- 04:00:57 - <Info> -- Loading distribution rule file /etc/suricata/rules/ipsec-events.rules
20/11/2025 -- 04:00:57 - <Info> -- Loading distribution rule file /etc/suricata/rules/kerberos-events.rules
20/11/2025 -- 04:00:57 - <Info> -- Loading distribution rule file /etc/suricata/rules/modbus-events.rules
20/11/2025 -- 04:00:57 - <Info> -- Loading distribution rule file /etc/suricata/rules/nfs-events.rules
20/11/2025 -- 04:00:57 - <Info> -- Loading distribution rule file /etc/suricata/rules/ntp-events.rules
20/11/2025 -- 04:00:57 - <Info> -- Loading distribution rule file /etc/suricata/rules/smb-events.rules
20/11/2025 -- 04:00:57 - <Info> -- Loading distribution rule file /etc/suricata/rules/smtp-events.rules
20/11/2025 -- 04:00:57 - <Info> -- Loading distribution rule file /etc/suricata/rules/stream-events.rules
20/11/2025 -- 04:00:57 - <Info> -- Loading distribution rule file /etc/suricata/rules/tls-events.rules
```

Elastic IP



Elastic IP address associated successfully.
Elastic IP address 34.206.186.51 has been associated with instance i-0c6113c4cee07ec5f

Elastic IP addresses (1/1) Info

Actions
Allocate Elastic IP address
Find elastic IP addresses by attribute or tag
Public IPv4 address : 34.206.186.51
Clear filters

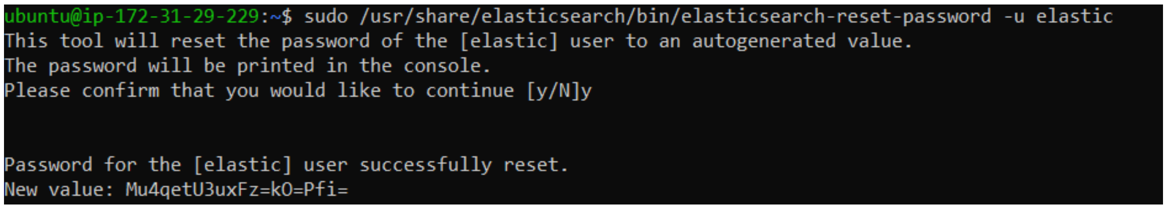
Name	Allocated IPv4 addr...	Type	Allocation ID	Reverse DNS record
elasticproject	34.206.186.51	Public IP	eipalloc-0432e09bce631acc8	–

34.206.186.51

Summary

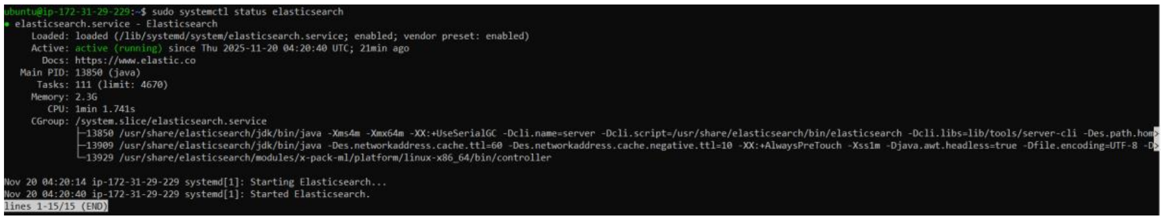
Allocated IPv4 address 34.206.186.51	**Type** Public IP	**Allocation ID** eipalloc-0432e09bce631acc8	**Reverse DNS record** –
Association ID eipassoc-003a8b0a98cda294b	**Scope** VPC	**Associated instance ID** i-0c6113c4cee07ec5f	**Private IP address** 172.31.29.229
Network interface ID eni-063ad2734acad885b	**Network interface owner account ID** 211125570159	**Public DNS** ec2-34-206-186-51.compute-1.amazonaws.com	**NAT Gateway ID** –
Address pool	**Network border group**	**Service managed**	

Elastic Password



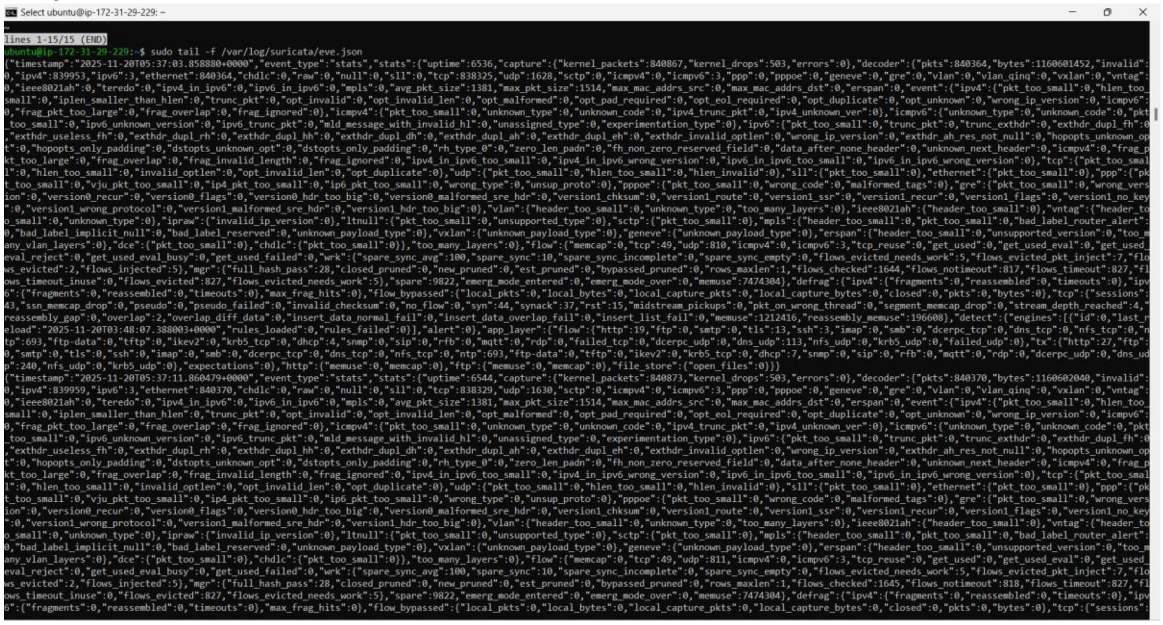
```
ubuntu@ip-172-31-29-229:~$ sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic
This tool will reset the password of the [elastic] user to an autogenerated value.
The password will be printed in the console.
Please confirm that you would like to continue [y/N]y


Password for the [elastic] user successfully reset.
New value: Mu4qetU3uxFz=kO=Pfi=
```

Elasticsearch



```
ubuntu@ip-172-31-29-229:~$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2025-11-20 04:20:40 UTC; 21min ago
       Docs: https://www.elastic.co
   Main PID: 13850 (java)
      Tasks: 111 (limit: 4670)
     Memory: 2.3G
        CPU: 1min 1.741s
     CGroup: /system.slice/elasticsearch.service
             ├─13850 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcli.name=server -Dcli.script=/usr/share/elasticsearch/bin/elasticsearch -Dcli.libs=lib/tools/server-cli -Des.path.hom>
             ├─13909 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreTouch -Xss1m -Djava.awt.headless=true -Dfile.encoding=UTF-8 -D>
             └─13929 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

Nov 20 04:20:14 ip-172-31-29-229 systemd[1]: Starting Elasticsearch...
Nov 20 04:20:40 ip-172-31-29-229 systemd[1]: Started Elasticsearch.
lines 1-15/15 (END)
```

Evejson

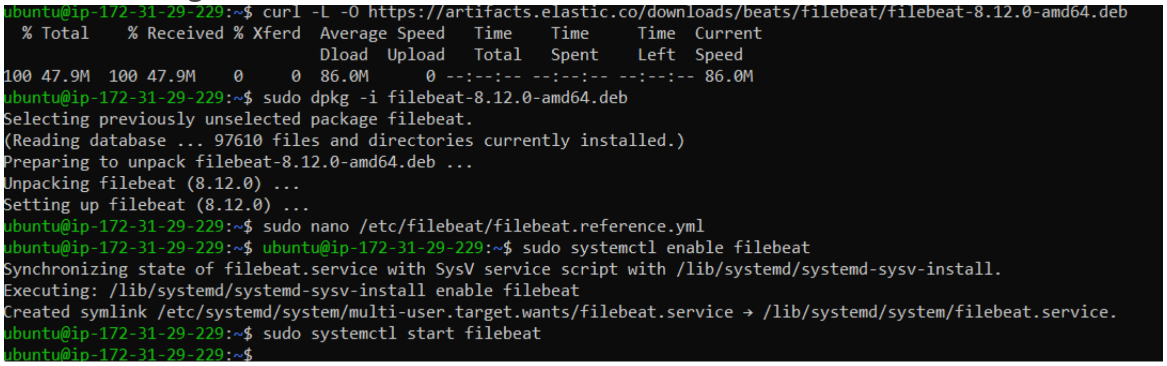


Select ubuntu@ip-172-31-29-229: ~

```
~
lines 1-15/15 (END)
ubuntu@ip-172-31-29-229:~$ sudo tail -f /var/log/suricata/eve.json
```

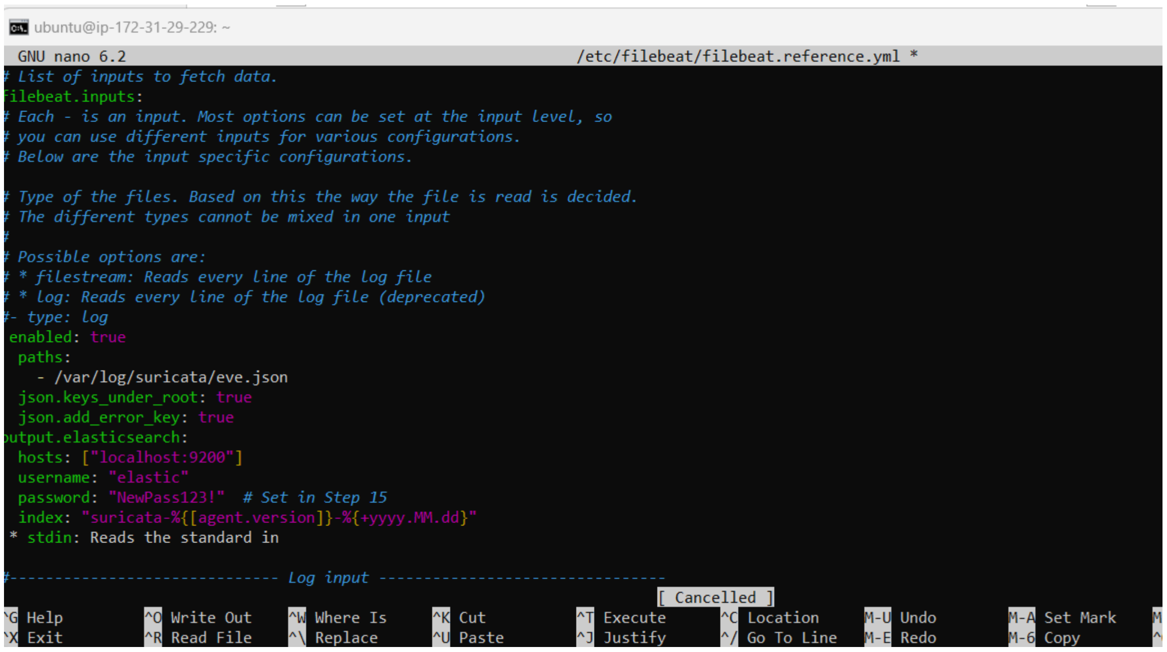
```
{"timestamp":"2025-11-20T05:37:03.858880+0000","event_type":"stats","stats":{"uptime":6536,"capture":{"kernel_packets":840867,"kernel_drops":503,"errors":0},"decoder":{"pkts":840364,"bytes":1160601452,"invalid":
0,"ipv4":839953,"ipv6":3,"ethernet":840364,"chdlc":0,"raw":0,"null":0,"sll":0,"tcp":838325,"udp":1628,"sctp":0,"icmpv4":0,"icmpv6":3,"ppp":0,"pppoe":0,"geneve":0,"gre":0,"vlan":0,"vlan_qinq":0,"vxlan":0,"vntag":
0,"ieee8021ah":0,"teredo":0,"ipv4_in_ipv6":0,"ipv6_in_ipv6":0,"mpls":0,"avg_pkt_size":1381,"max_pkt_size":1514,"max_mac_addrs_src":0,"max_mac_addrs_dst":0,"erspan":0,"event":{"ipv4":{"pkt_too_small":0,"hlen_too_
small":0,"iplen_smaller_than_hlen":0,"trunc_pkt":0,"opt_invalid":0,"opt_invalid_len":0,"opt_malformed":0,"opt_pad_required":0,"opt_eol_required":0,"opt_duplicate":0,"opt_unknown":0,"wrong_ip_version":0,"icmpv6":
0,"frag_pkt_too_large":0,"frag_overlap":0,"frag_ignored":0},"icmpv4":{"pkt_too_small":0,"unknown_type":0,"unknown_code":0,"ipv4_trunc_pkt":0,"ipv4_unknown_ver":0},"icmpv6":{"unknown_type":0,"unknown_code":0,"pkt
_too_small":0,"ipv6_unknown_version":0,"ipv6_trunc_pkt":0,"mld_message_with_invalid_hl":0,"unassigned_type":0,"experimentation_type":0},"ipv6":{"pkt_too_small":0,"trunc_pkt":0,"trunc_exthdr":0,"exthdr_dupl_fh":0
,"exthdr_useless_fh":0,"exthdr_dupl_rh":0,"exthdr_dupl_hh":0,"exthdr_dupl_dh":0,"exthdr_dupl_ah":0,"exthdr_dupl_eh":0,"exthdr_invalid_optlen":0,"wrong_ip_version":0,"exthdr_ah_res_not_null":0,"hopopts_unknown_op
t":0,"hopopts_only_padding":0,"dstopts_unknown_opt":0,"dstopts_only_padding":0,"rh_type_0":0,"zero_len_padn":0,"fh_non_zero_reserved_field":0,"data_after_none_header":0,"unknown_next_header":0,"icmpv4":0,"frag_p
kt_too_large":0,"frag_overlap":0,"frag_invalid_length":0,"frag_ignored":0,"ipv4_in_ipv6_too_small":0,"ipv4_in_ipv6_wrong_version":0,"ipv6_in_ipv6_too_small":0,"ipv6_in_ipv6_wrong_version":0},"tcp":{"pkt_too_smal
l":0,"hlen_too_small":0,"invalid_optlen":0,"opt_invalid_len":0,"opt_duplicate":0},"udp":{"pkt_too_small":0,"hlen_too_small":0,"hlen_invalid":0},"sll":{"pkt_too_small":0},"ethernet":{"pkt_too_small":0},"ppp":{"pk
t_too_small":0,"vju_pkt_too_small":0,"ip4_pkt_too_small":0,"ip6_pkt_too_small":0,"wrong_type":0,"unsup_proto":0},"pppoe":{"pkt_too_small":0,"wrong_code":0,"malformed_tags":0},"gre":{"pkt_too_small":0,"wrong_vers
ion":0,"version0_recur":0,"version0_flags":0,"version0_hdr_too_big":0,"version0_malformed_sre_hdr":0,"version1_chksum":0,"version1_route":0,"version1_ssr":0,"version1_recur":0,"version1_flags":0,"version1_no_key
":0,"version1_wrong_protocol":0,"version1_malformed_sre_hdr":0,"version1_hdr_too_big":0},"vlan":{"header_too_small":0,"unknown_type":0,"too_many_layers":0},"ieee8021ah":{"header_too_small":0},"vntag":{"header_to
o_small":0,"unknown_type":0},"ipraw":{"invalid_ip_version":0},"ltnull":{"pkt_too_small":0,"unsupported_type":0},"sctp":{"pkt_too_small":0},"mpls":{"header_too_small":0,"pkt_too_small":0,"bad_label_router_alert":
0,"bad_label_implicit_null":0,"bad_label_reserved":0,"unknown_payload_type":0},"vxlan":{"unknown_payload_type":0},"geneve":{"unknown_payload_type":0},"erspan":{"header_too_small":0,"unsupported_version":0,"too_m
any_vlan_layers":0},"dce":{"pkt_too_small":0},"chdlc":{"pkt_too_small":0}},"too_many_layers":0},"flow":{"memcap":0,"tcp":49,"udp":810,"icmpv4":0,"icmpv6":3,"tcp_reuse":0,"get_used":0,"get_used_eval":0,"get_used_
eval_reject":0,"get_used_eval_busy":0,"get_used_failed":0,"wrk":{"spare_sync_avg":100,"spare_sync":10,"spare_sync_incomplete":0,"spare_sync_empty":0,"flows_evicted_needs_work":5,"flows_evicted_pkt_inject":7,"flo
ws_evicted":2,"flows_injected":5},"mgr":{"full_hash_pass":28,"closed_pruned":0,"new_pruned":0,"est_pruned":0,"bypassed_pruned":0,"rows_maxlen":1,"flows_checked":1644,"flows_notimeout":817,"flows_timeout":827,"fl
ows_timeout_inuse":0,"flows_evicted":827,"flows_evicted_needs_work":5},"spare":9822,"emerg_mode_entered":0,"emerg_mode_over":0,"memuse":7474304},"defrag":{"ipv4":{"fragments":0,"reassembled":0,"timeouts":0},"ipv
6":{"fragments":0,"reassembled":0,"timeouts":0},"max_frag_hits":0},"flow_bypassed":{"local_pkts":0,"local_bytes":0,"local_capture_pkts":0,"local_capture_bytes":0,"closed":0,"pkts":0,"bytes":0},"tcp":{"sessions":
43,"ssn_memcap_drop":0,"pseudo":0,"pseudo_failed":0,"invalid_checksum":0,"no_flow":0,"syn":44,"synack":37,"rst":15,"midstream_pickups":0,"pkt_on_wrong_thread":0,"segment_memcap_drop":0,"stream_depth_reached":4,"
reassembly_gap":0,"overlap":2,"overlap_diff_data":0,"insert_data_normal_fail":0,"insert_data_overlap_fail":0,"insert_list_fail":0,"memuse":1212416,"reassembly_memuse":196608},"detect":{"engines":[{"id":0,"last_r
eload":"2025-11-20T03:48:07.388003+0000","rules_loaded":0,"rules_failed":0}],"alert":0},"app_layer":{"flow":{"http":19,"ftp":0,"smtp":0,"tls":13,"ssh":3,"imap":0,"smb":0,"dcerpc_tcp":0,"dns_tcp":0,"nfs_tcp":0,"n
tp":693,"ftp-data":0,"tftp":0,"ikev2":0,"krb5_tcp":0,"dhcp":4,"snmp":0,"sip":0,"rfb":0,"mqtt":0,"rdp":0,"failed_tcp":0,"dcerpc_udp":0,"dns_udp":113,"nfs_udp":0,"krb5_udp":0,"failed_udp":0},"tx":{"http":27,"ftp":
0,"smtp":0,"tls":0,"ssh":0,"imap":0,"smb":0,"dcerpc_tcp":0,"dns_tcp":0,"nfs_tcp":0,"ntp":693,"ftp-data":0,"tftp":0,"ikev2":0,"krb5_tcp":0,"dhcp":7,"snmp":0,"sip":0,"rfb":0,"mqtt":0,"rdp":0,"dcerpc_udp":0,"dns_ud
p":240,"nfs_udp":0,"krb5_udp":0},"expectations":0},"http":{"memuse":0,"memcap":0},"ftp":{"memuse":0,"memcap":0},"file_store":{"open_files":0}}}
{"timestamp":"2025-11-20T05:37:11.860479+0000","event_type":"stats","stats":{"uptime":6544,"capture":{"kernel_packets":840873,"kernel_drops":503,"errors":0},"decoder":{"pkts":840370,"bytes":1160602040,"invalid":
0,"ipv4":839959,"ipv6":3,"ethernet":840370,"chdlc":0,"raw":0,"null":0,"sll":0,"tcp":838329,"udp":1630,"sctp":0,"icmpv4":0,"icmpv6":3,"ppp":0,"pppoe":0,"geneve":0,"gre":0,"vlan":0,"vlan_qinq":0,"vxlan":0,"vntag":
0,"ieee8021ah":0,"teredo":0,"ipv4_in_ipv6":0,"ipv6_in_ipv6":0,"mpls":0,"avg_pkt_size":1381,"max_pkt_size":1514,"max_mac_addrs_src":0,"max_mac_addrs_dst":0,"erspan":0,"event":{"ipv4":{"pkt_too_small":0,"hlen_too_
small":0,"iplen_smaller_than_hlen":0,"trunc_pkt":0,"opt_invalid":0,"opt_invalid_len":0,"opt_malformed":0,"opt_pad_required":0,"opt_eol_required":0,"opt_duplicate":0,"opt_unknown":0,"wrong_ip_version":0,"icmpv6":
0,"frag_pkt_too_large":0,"frag_overlap":0,"frag_ignored":0},"icmpv4":{"pkt_too_small":0,"unknown_type":0,"unknown_code":0,"ipv4_trunc_pkt":0,"ipv4_unknown_ver":0},"icmpv6":{"unknown_type":0,"unknown_code":0,"pkt
_too_small":0,"ipv6_unknown_version":0,"ipv6_trunc_pkt":0,"mld_message_with_invalid_hl":0,"unassigned_type":0,"experimentation_type":0},"ipv6":{"pkt_too_small":0,"trunc_pkt":0,"trunc_exthdr":0,"exthdr_dupl_fh":0
,"exthdr_useless_fh":0,"exthdr_dupl_rh":0,"exthdr_dupl_hh":0,"exthdr_dupl_dh":0,"exthdr_dupl_ah":0,"exthdr_dupl_eh":0,"exthdr_invalid_optlen":0,"wrong_ip_version":0,"exthdr_ah_res_not_null":0,"hopopts_unknown_op
t":0,"hopopts_only_padding":0,"dstopts_unknown_opt":0,"dstopts_only_padding":0,"rh_type_0":0,"zero_len_padn":0,"fh_non_zero_reserved_field":0,"data_after_none_header":0,"unknown_next_header":0,"icmpv4":0,"frag_p
kt_too_large":0,"frag_overlap":0,"frag_invalid_length":0,"frag_ignored":0,"ipv4_in_ipv6_too_small":0,"ipv4_in_ipv6_wrong_version":0,"ipv6_in_ipv6_too_small":0,"ipv6_in_ipv6_wrong_version":0},"tcp":{"pkt_too_smal
l":0,"hlen_too_small":0,"invalid_optlen":0,"opt_invalid_len":0,"opt_duplicate":0},"udp":{"pkt_too_small":0,"hlen_too_small":0,"hlen_invalid":0},"sll":{"pkt_too_small":0},"ethernet":{"pkt_too_small":0},"ppp":{"pk
t_too_small":0,"vju_pkt_too_small":0,"ip4_pkt_too_small":0,"ip6_pkt_too_small":0,"wrong_type":0,"unsup_proto":0},"pppoe":{"pkt_too_small":0,"wrong_code":0,"malformed_tags":0},"gre":{"pkt_too_small":0,"wrong_vers
ion":0,"version0_recur":0,"version0_flags":0,"version0_hdr_too_big":0,"version0_malformed_sre_hdr":0,"version1_chksum":0,"version1_route":0,"version1_ssr":0,"version1_recur":0,"version1_flags":0,"version1_no_key
":0,"version1_wrong_protocol":0,"version1_malformed_sre_hdr":0,"version1_hdr_too_big":0},"vlan":{"header_too_small":0,"unknown_type":0,"too_many_layers":0},"ieee8021ah":{"header_too_small":0},"vntag":{"header_to
o_small":0,"unknown_type":0},"ipraw":{"invalid_ip_version":0},"ltnull":{"pkt_too_small":0,"unsupported_type":0},"sctp":{"pkt_too_small":0},"mpls":{"header_too_small":0,"pkt_too_small":0,"bad_label_router_alert":
0,"bad_label_implicit_null":0,"bad_label_reserved":0,"unknown_payload_type":0},"vxlan":{"unknown_payload_type":0},"geneve":{"unknown_payload_type":0},"erspan":{"header_too_small":0,"unsupported_version":0,"too_m
any_vlan_layers":0},"dce":{"pkt_too_small":0},"chdlc":{"pkt_too_small":0}},"too_many_layers":0},"flow":{"memcap":0,"tcp":49,"udp":811,"icmpv4":0,"icmpv6":3,"tcp_reuse":0,"get_used":0,"get_used_eval":0,"get_used_
eval_reject":0,"get_used_eval_busy":0,"get_used_failed":0,"wrk":{"spare_sync_avg":100,"spare_sync":10,"spare_sync_incomplete":0,"spare_sync_empty":0,"flows_evicted_needs_work":5,"flows_evicted_pkt_inject":7,"flo
ws_evicted":2,"flows_injected":5},"mgr":{"full_hash_pass":28,"closed_pruned":0,"new_pruned":0,"est_pruned":0,"bypassed_pruned":0,"rows_maxlen":1,"flows_checked":1645,"flows_notimeout":818,"flows_timeout":827,"fl
ows_timeout_inuse":0,"flows_evicted":827,"flows_evicted_needs_work":5},"spare":9822,"emerg_mode_entered":0,"emerg_mode_over":0,"memuse":7474304},"defrag":{"ipv4":{"fragments":0,"reassembled":0,"timeouts":0},"ipv
6":{"fragments":0,"reassembled":0,"timeouts":0},"max_frag_hits":0},"flow_bypassed":{"local_pkts":0,"local_bytes":0,"local_capture_pkts":0,"local_capture_bytes":0,"closed":0,"pkts":0,"bytes":0},"tcp":{"sessions":
```

Filebeatconfig



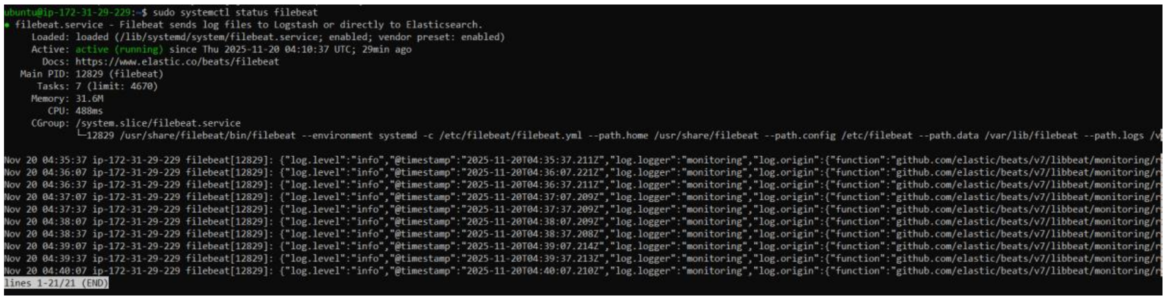
```
ubuntu@ip-172-31-29-229:~$ curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-8.12.0-amd64.deb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 47.9M  100 47.9M    0     0  86.0M      0 --:--:-- --:--:-- --:--:-- 86.0M
ubuntu@ip-172-31-29-229:~$ sudo dpkg -i filebeat-8.12.0-amd64.deb
Selecting previously unselected package filebeat.
(Reading database ... 97610 files and directories currently installed.)
Preparing to unpack filebeat-8.12.0-amd64.deb ...
Unpacking filebeat (8.12.0) ...
Setting up filebeat (8.12.0) ...
ubuntu@ip-172-31-29-229:~$ sudo nano /etc/filebeat/filebeat.reference.yml
ubuntu@ip-172-31-29-229:~$ ubuntu@ip-172-31-29-229:~$ sudo systemctl enable filebeat
Synchronizing state of filebeat.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable filebeat
Created symlink /etc/systemd/system/multi-user.target.wants/filebeat.service → /lib/systemd/system/filebeat.service.
ubuntu@ip-172-31-29-229:~$ sudo systemctl start filebeat
ubuntu@ip-172-31-29-229:~$
```

Filebeatref



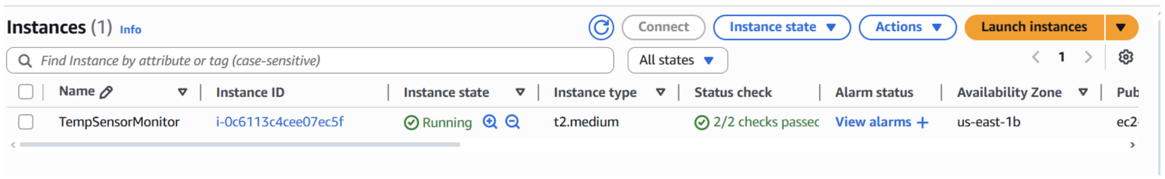
```
ubuntu@ip-172-31-29-229: ~
 GNU nano 6.2                                  /etc/filebeat/filebeat.reference.yml *
# List of inputs to fetch data.
filebeat.inputs:
# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input specific configurations.

# Type of the files. Based on this the way the file is read is decided.
# The different types cannot be mixed in one input
#
# Possible options are:
# * filestream: Reads every line of the log file
# * log: Reads every line of the log file (deprecated)
#- type: log
 enabled: true
  paths:
    - /var/log/suricata/eve.json
  json.keys_under_root: true
  json.add_error_key: true
output.elasticsearch:
  hosts: ["localhost:9200"]
  username: "elastic"
  password: "NewPass123!"  # Set in Step 15
  index: "suricata-%{[agent.version]}-%{+yyyy.MM.dd}"
 * stdin: Reads the standard in

#---------------------------- Log input -------------------------------
                                                 [ Cancelled ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^
```

Filebeatstat



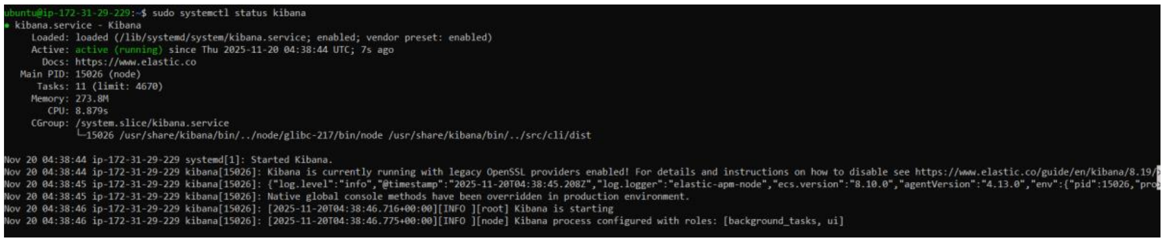
```
ubuntu@ip-172-31-29-229:~$ sudo systemctl status filebeat
● filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
     Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2025-11-20 04:10:37 UTC; 29min ago
       Docs: https://www.elastic.co/beats/filebeat
   Main PID: 12829 (filebeat)
      Tasks: 7 (limit: 4670)
     Memory: 31.6M
        CPU: 488ms
     CGroup: /system.slice/filebeat.service
             └─12829 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc/filebeat/filebeat.yml --path.home /usr/share/filebeat --path.config /etc/filebeat --path.data /var/lib/filebeat --path.logs /v

Nov 20 04:35:37 ip-172-31-29-229 filebeat[12829]: {"log.level":"info","@timestamp":"2025-11-20T04:35:37.211Z","log.logger":"monitoring","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/monitoring/r
Nov 20 04:36:07 ip-172-31-29-229 filebeat[12829]: {"log.level":"info","@timestamp":"2025-11-20T04:36:07.221Z","log.logger":"monitoring","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/monitoring/r
Nov 20 04:36:37 ip-172-31-29-229 filebeat[12829]: {"log.level":"info","@timestamp":"2025-11-20T04:36:37.211Z","log.logger":"monitoring","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/monitoring/r
Nov 20 04:37:07 ip-172-31-29-229 filebeat[12829]: {"log.level":"info","@timestamp":"2025-11-20T04:37:07.209Z","log.logger":"monitoring","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/monitoring/r
Nov 20 04:37:37 ip-172-31-29-229 filebeat[12829]: {"log.level":"info","@timestamp":"2025-11-20T04:37:37.209Z","log.logger":"monitoring","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/monitoring/r
Nov 20 04:38:07 ip-172-31-29-229 filebeat[12829]: {"log.level":"info","@timestamp":"2025-11-20T04:38:07.209Z","log.logger":"monitoring","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/monitoring/r
Nov 20 04:38:37 ip-172-31-29-229 filebeat[12829]: {"log.level":"info","@timestamp":"2025-11-20T04:38:37.208Z","log.logger":"monitoring","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/monitoring/r
Nov 20 04:39:07 ip-172-31-29-229 filebeat[12829]: {"log.level":"info","@timestamp":"2025-11-20T04:39:07.214Z","log.logger":"monitoring","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/monitoring/r
Nov 20 04:39:37 ip-172-31-29-229 filebeat[12829]: {"log.level":"info","@timestamp":"2025-11-20T04:39:37.213Z","log.logger":"monitoring","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/monitoring/r
Nov 20 04:40:07 ip-172-31-29-229 filebeat[12829]: {"log.level":"info","@timestamp":"2025-11-20T04:40:07.210Z","log.logger":"monitoring","log.origin":{"function":"github.com/elastic/beats/v7/libbeat/monitoring/r
lines 1-21/21 (END)
```

Instance



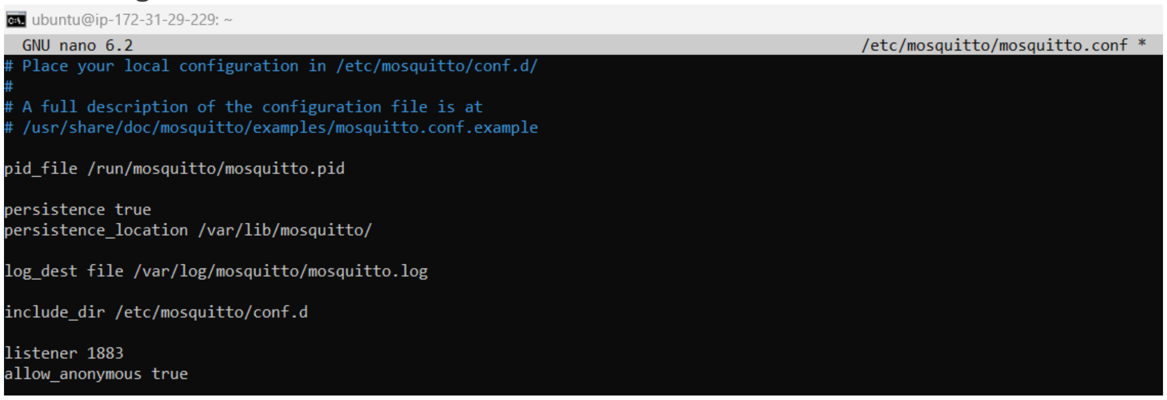
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pub
☐	TempSensorMonitor	i-0c6113c4cee07ec5f	✓ Running	t2.medium	✓ 2/2 checks passec	View alarms +	us-east-1b	ec2-

Kibanastat



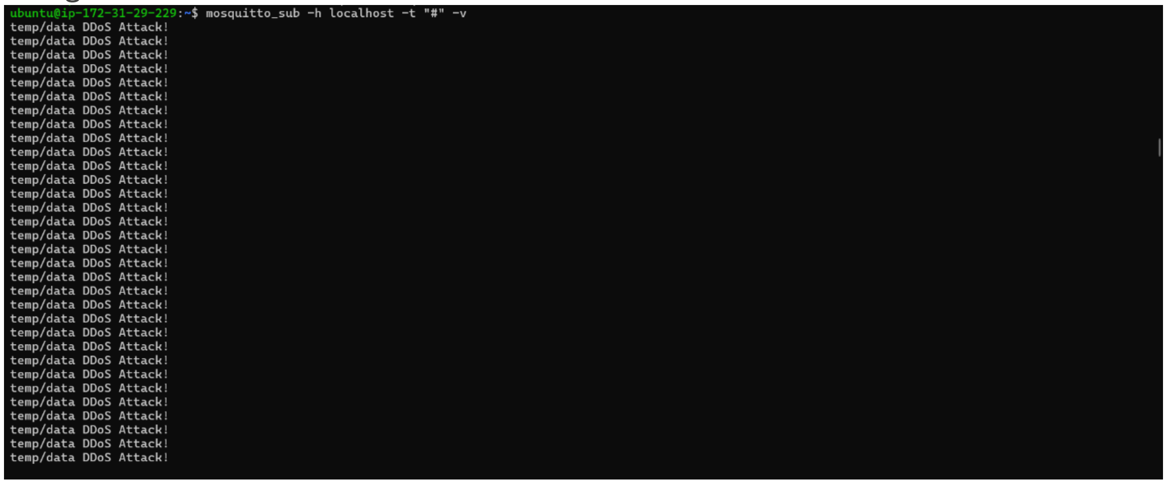
```
ubuntu@ip-172-31-29-229:~$ sudo systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/lib/systemd/system/kibana.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2025-11-20 04:38:44 UTC; 7s ago
       Docs: https://www.elastic.co
   Main PID: 15026 (node)
      Tasks: 11 (limit: 4670)
     Memory: 273.8M
        CPU: 8.879s
     CGroup: /system.slice/kibana.service
             └─15026 /usr/share/kibana/bin/../node/glibc-217/bin/node /usr/share/kibana/bin/../src/cli/dist

Nov 20 04:38:44 ip-172-31-29-229 systemd[1]: Started Kibana.
Nov 20 04:38:44 ip-172-31-29-229 kibana[15026]: Kibana is currently running with legacy OpenSSL providers enabled! For details and instructions on how to disable see https://www.elastic.co/guide/en/kibana/8.19/
Nov 20 04:38:45 ip-172-31-29-229 kibana[15026]: {"log.level":"info","@timestamp":"2025-11-20T04:38:45.208Z","log.logger":"elastic-apm-node","ecs.version":"8.10.0","agentVersion":"4.13.0","env":{"pid":15026,"pro
Nov 20 04:38:45 ip-172-31-29-229 kibana[15026]: Native global console methods have been overridden in production environment.
Nov 20 04:38:46 ip-172-31-29-229 kibana[15026]: [2025-11-20T04:38:46.716+00:00][INFO ][root] Kibana is starting
Nov 20 04:38:46 ip-172-31-29-229 kibana[15026]: [2025-11-20T04:38:46.775+00:00][INFO ][node] Kibana process configured with roles: [background_tasks, ui]
```

Mosconfig



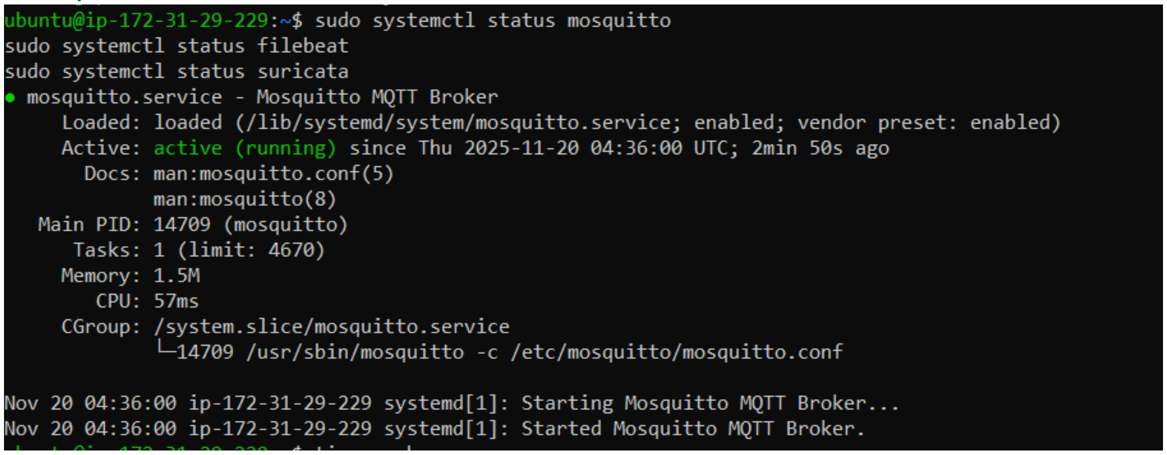
```
ubuntu@ip-172-31-29-229: ~
  GNU nano 6.2                                                        /etc/mosquitto/mosquitto.conf *
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

pid_file /run/mosquitto/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d

listener 1883
allow_anonymous true
```

Mosgotattacked



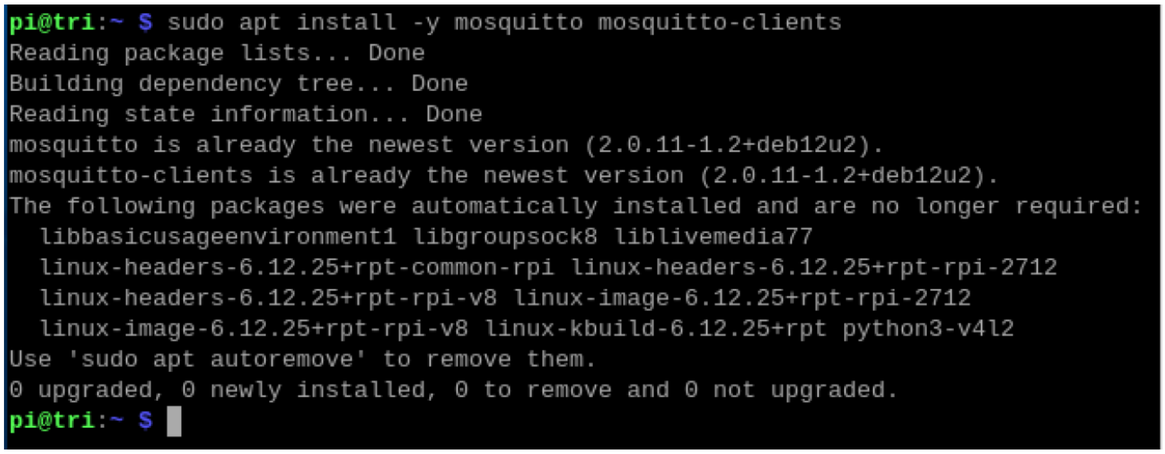
```
ubuntu@ip-172-31-29-229:~$ mosquitto_sub -h localhost -t "#" -v
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
temp/data DDoS Attack!
```

Mosquittoinstall



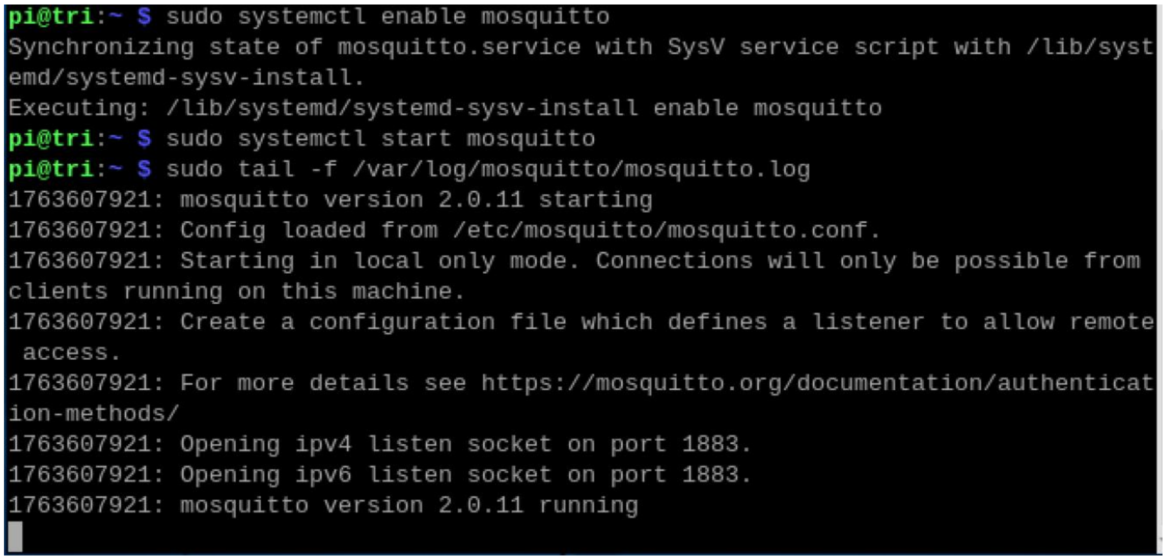
```
ubuntu@ip-172-31-29-229:~$ sudo systemctl status mosquitto
sudo systemctl status filebeat
sudo systemctl status suricata
● mosquitto.service - Mosquitto MQTT Broker
     Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2025-11-20 04:36:00 UTC; 2min 50s ago
       Docs: man:mosquitto.conf(5)
             man:mosquitto(8)
   Main PID: 14709 (mosquitto)
      Tasks: 1 (limit: 4670)
     Memory: 1.5M
        CPU: 57ms
     CGroup: /system.slice/mosquitto.service
             └─14709 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

Nov 20 04:36:00 ip-172-31-29-229 systemd[1]: Starting Mosquitto MQTT Broker...
Nov 20 04:36:00 ip-172-31-29-229 systemd[1]: Started Mosquitto MQTT Broker.
```

Mosstat



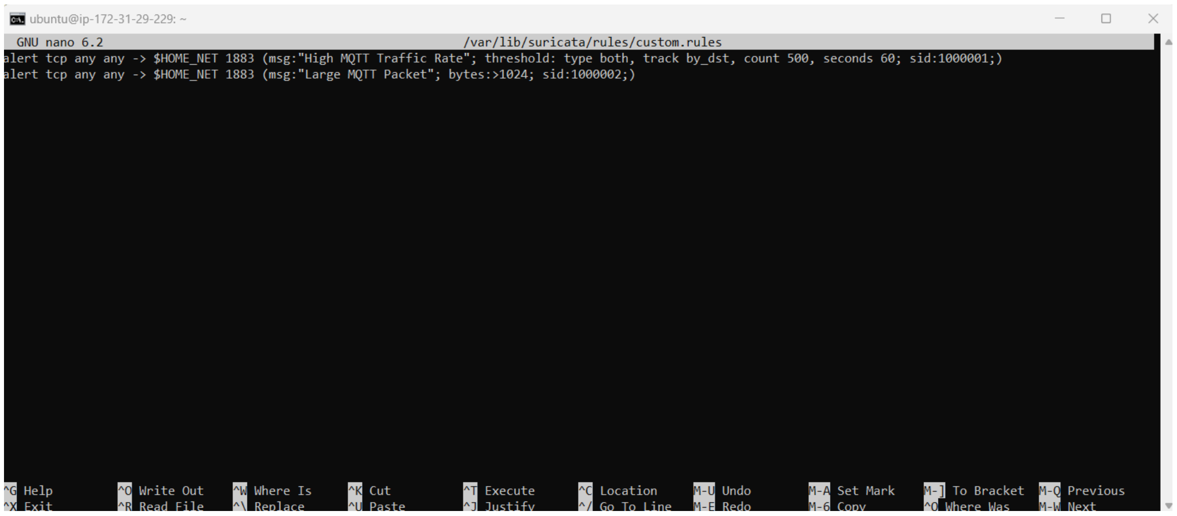
```
pi@tri:~ $ sudo apt install -y mosquitto mosquitto-clients
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mosquitto is already the newest version (2.0.11-1.2+deb12u2).
mosquitto-clients is already the newest version (2.0.11-1.2+deb12u2).
The following packages were automatically installed and are no longer required:
  libbasicusageenvironment1 libgroupsock8 liblivemedia77
  linux-headers-6.12.25+rpt-common-rpi linux-headers-6.12.25+rpt-rpi-2712
  linux-headers-6.12.25+rpt-rpi-v8 linux-image-6.12.25+rpt-rpi-2712
  linux-image-6.12.25+rpt-rpi-v8 linux-kbuild-6.12.25+rpt python3-v4l2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@tri:~ $
```

Mosstatus



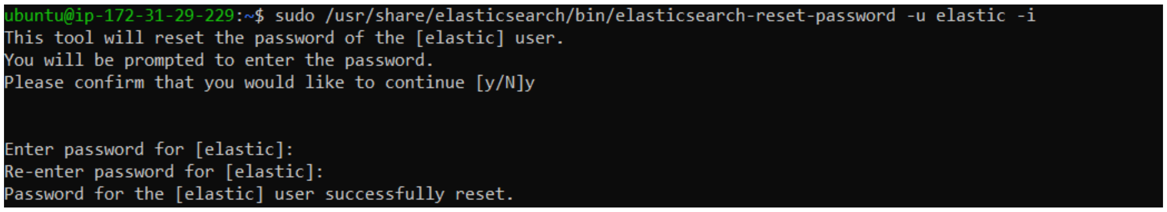
```
pi@tri:~ $ sudo systemctl enable mosquitto
Synchronizing state of mosquitto.service with SysV service script with /lib/syst
emd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mosquitto
pi@tri:~ $ sudo systemctl start mosquitto
pi@tri:~ $ sudo tail -f /var/log/mosquitto/mosquitto.log
1763607921: mosquitto version 2.0.11 starting
1763607921: Config loaded from /etc/mosquitto/mosquitto.conf.
1763607921: Starting in local only mode. Connections will only be possible from
clients running on this machine.
1763607921: Create a configuration file which defines a listener to allow remote
 access.
1763607921: For more details see https://mosquitto.org/documentation/authenticat
ion-methods/
1763607921: Opening ipv4 listen socket on port 1883.
1763607921: Opening ipv6 listen socket on port 1883.
1763607921: mosquitto version 2.0.11 running
```

Nanocustom



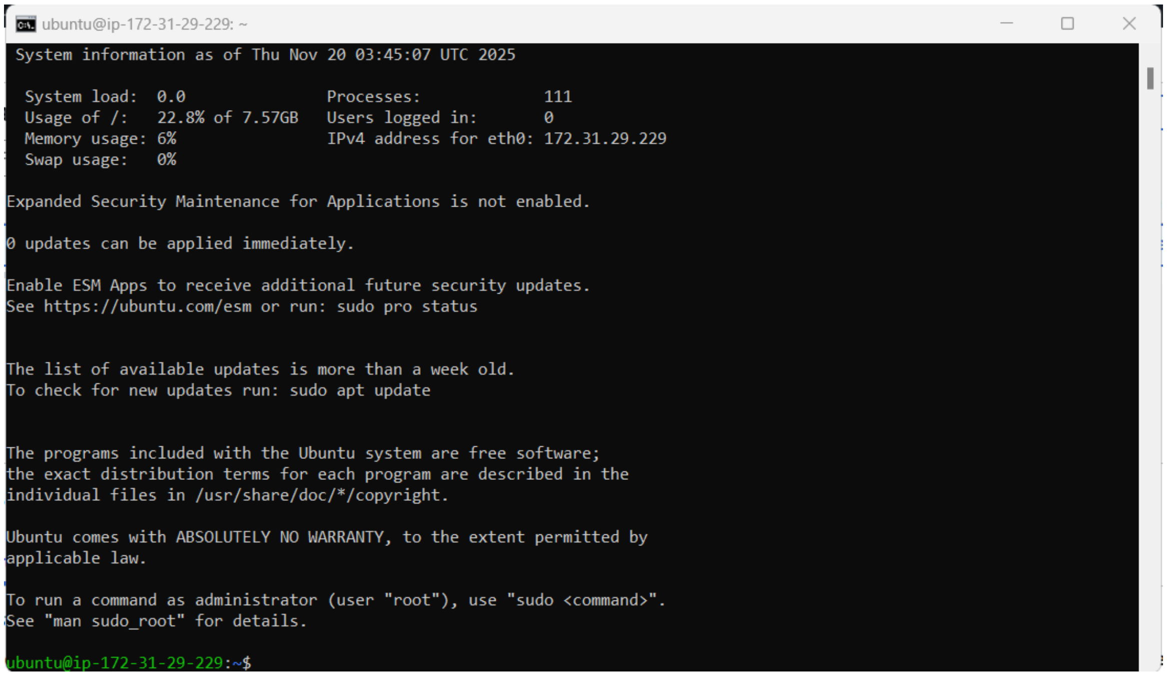
```
ubuntu@ip-172-31-29-229: ~
 GNU nano 6.2                                  /var/lib/suricata/rules/custom.rules
alert tcp any any -> $HOME_NET 1883 (msg:"High MQTT Traffic Rate"; threshold: type both, track by_dst, count 500, seconds 60; sid:1000001;)
alert tcp any any -> $HOME_NET 1883 (msg:"Large MQTT Packet"; bytes:>1024; sid:1000002;)

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket M-Q Previous
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was  M-W Next
```

NewPass123!



```
ubuntu@ip-172-31-29-229:~$ sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic -i
This tool will reset the password of the [elastic] user.
You will be prompted to enter the password.
Please confirm that you would like to continue [y/N]y


Enter password for [elastic]:
Re-enter password for [elastic]:
Password for the [elastic] user successfully reset.
```

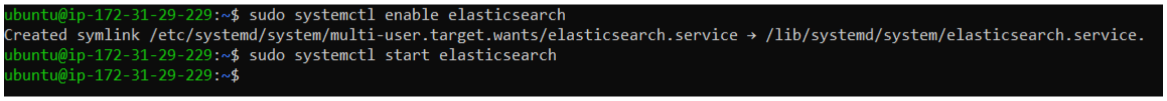
SSH



ubuntu@ip-172-31-29-229: ~

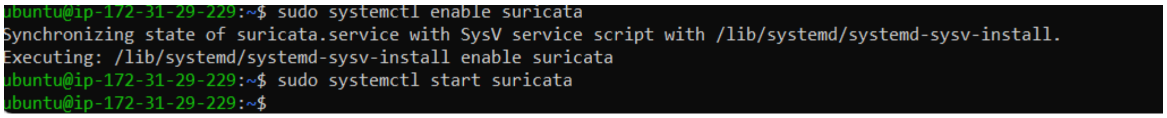
```
 System information as of Thu Nov 20 03:45:07 UTC 2025

  System load:  0.0               Processes:             111
  Usage of /:   22.8% of 7.57GB   Users logged in:       0
  Memory usage: 6%                IPv4 address for eth0: 172.31.29.229
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-29-229:~$
```

Startelasticsearch



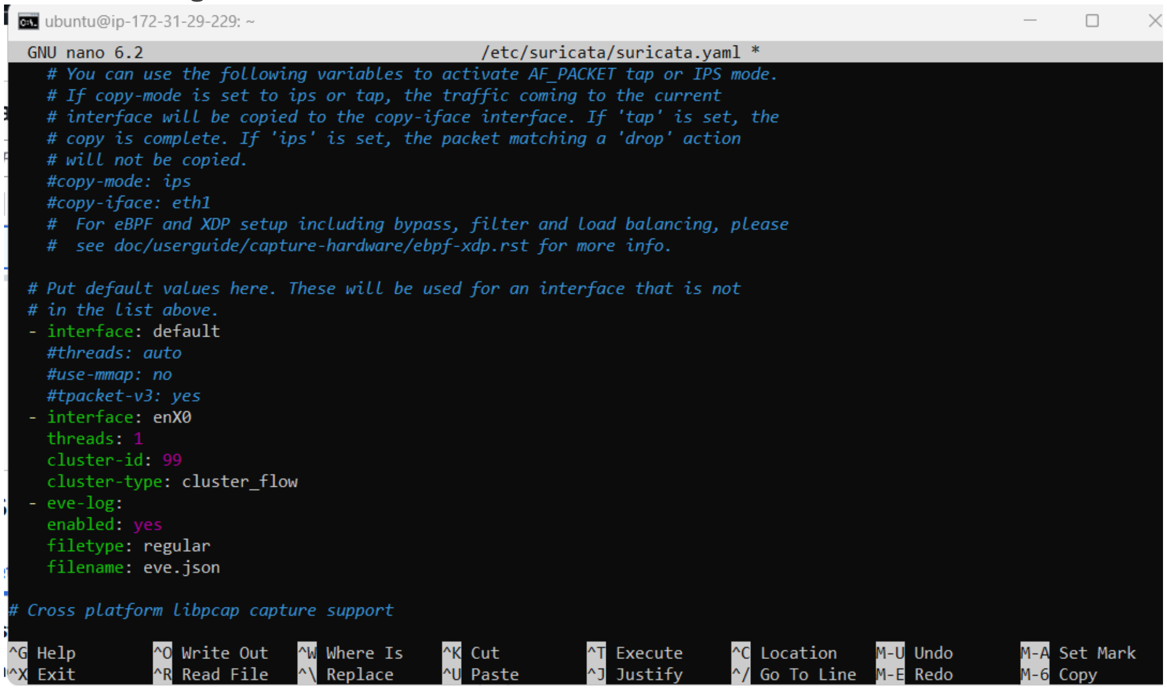
```
ubuntu@ip-172-31-29-229:~$ sudo systemctl enable elasticsearch
Created symlink /etc/systemd/system/multi-user.target.wants/elasticsearch.service → /lib/systemd/system/elasticsearch.service.
ubuntu@ip-172-31-29-229:~$ sudo systemctl start elasticsearch
ubuntu@ip-172-31-29-229:~$
```

Startsuricata



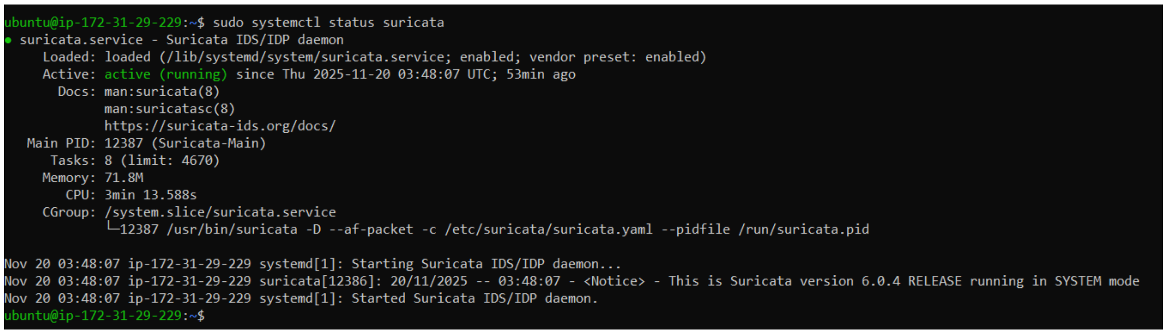
```
ubuntu@ip-172-31-29-229:~$ sudo systemctl enable suricata
Synchronizing state of suricata.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable suricata
ubuntu@ip-172-31-29-229:~$ sudo systemctl start suricata
ubuntu@ip-172-31-29-229:~$
```

Suricataconfig



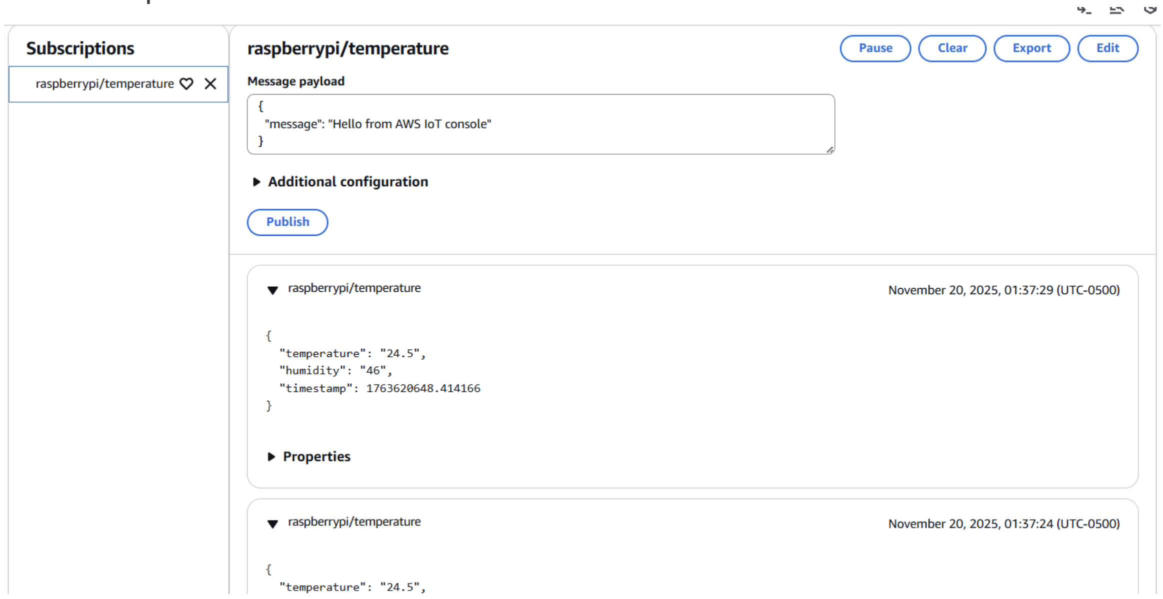
```
ubuntu@ip-172-31-29-229: ~
GNU nano 6.2                                     /etc/suricata/suricata.yaml *
    # You can use the following variables to activate AF_PACKET tap or IPS mode.
    # If copy-mode is set to ips or tap, the traffic coming to the current
    # interface will be copied to the copy-iface interface. If 'tap' is set, the
    # copy is complete. If 'ips' is set, the packet matching a 'drop' action
    # will not be copied.
    #copy-mode: ips
    #copy-iface: eth1
    #  For eBPF and XDP setup including bypass, filter and load balancing, please
    #  see doc/userguide/capture-hardware/ebpf-xdp.rst for more info.

  # Put default values here. These will be used for an interface that is not
  # in the list above.
  - interface: default
    #threads: auto
    #use-mmap: no
    #tpacket-v3: yes
  - interface: enX0
    threads: 1
    cluster-id: 99
    cluster-type: cluster_flow
  - eve-log:
    enabled: yes
    filetype: regular
    filename: eve.json

# Cross platform libpcap capture support

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy
```

Suricatastat



```
ubuntu@ip-172-31-29-229:~$ sudo systemctl status suricata
● suricata.service - Suricata IDS/IDP daemon
     Loaded: loaded (/lib/systemd/system/suricata.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2025-11-20 03:48:07 UTC; 53min ago
       Docs: man:suricata(8)
             man:suricatasc(8)
             https://suricata-ids.org/docs/
   Main PID: 12387 (Suricata-Main)
      Tasks: 8 (limit: 4670)
     Memory: 71.8M
        CPU: 3min 13.588s
     CGroup: /system.slice/suricata.service
             └─12387 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid

Nov 20 03:48:07 ip-172-31-29-229 systemd[1]: Starting Suricata IDS/IDP daemon...
Nov 20 03:48:07 ip-172-31-29-229 suricata[12386]: 20/11/2025 -- 03:48:07 - <Notice> - This is Suricata version 6.0.4 RELEASE running in SYSTEM mode
Nov 20 03:48:07 ip-172-31-29-229 systemd[1]: Started Suricata IDS/IDP daemon.
ubuntu@ip-172-31-29-229:~$
```

Timestamp



```
{
  "message": "Hello from AWS IoT console"
}
```

November 20, 2025, 01:37:29 (UTC-0500)

```
{
  "temperature": "24.5",
  "humidity": "46",
  "timestamp": 1763620648.414166
}
```

November 20, 2025, 01:37:24 (UTC-0500)

```
{
  "temperature": "24.5",
```