

# Detailed Guide: Setting Up Logstash with Persistent Queue on Windows and Forwarding Logs to Wazuh on Ubuntu

## Goal

- **Logstash (Windows 11 VM)** stores logs **on disk** if disconnected from Wazuh.
  - When **reconnected**, Logstash forwards all stored logs **to Wazuh**.
  - Logs are stored persistently in **C:\logstash\data\queue**.
- 

## ◆ Step 1: Install and Configure Logstash on Windows 11 VM

Logstash will collect logs, store them **on disk** when disconnected, and forward them when reconnected.

### 1. Install Logstash

1. Download Logstash from the official Elastic site:  
[Download Logstash](#)
2. Extract the ZIP file to C:\logstash.
3. Set up the Logstash configuration folder:

```
powershell  
CopyEdit  
mkdir C:\logstash\config
```

---

### 2. Configure Logstash for Persistent Queues

Edit the main **Logstash settings file**:

```
powershell  
CopyEdit  
notepad C:\logstash\config\logstash.yml
```

Modify or add the following:

```
yaml  
CopyEdit  
# Enable Persistent Queue (PQ) to store logs on disk  
queue.type: persisted  
queue.max_bytes: 10gb      # Store up to 10GB of logs before overwriting  
queue.checkpoint.acks: 1  
queue.checkpoint.writes: 1
```

```
queue.drain: true          # Ensures logs are forwarded once Wazuh is back online

# Define where Logstash stores persistent logs
path.data: C:/logstash/data
```

Save and exit.

---

### 3. Configure Logstash Pipeline to Store and Forward Logs

Create a new Logstash pipeline file:

```
powershell
CopyEdit
notepad C:\logstash\config\logstash.conf
```

Paste the following:

```
yaml
CopyEdit
input {
    file {
        path => "C:/logs/*.log"
        start_position => "beginning"
        since_db_path => "NUL"
    }
}

filter {
    mutate {
        add_field => { "logsource" => "logstash_windows" }
    }
}

output {
    # Store logs locally when disconnected
    file {
        path => "C:/logstash_queue/logstash_backup.log"
        codec => "json"
    }
}

# Forward logs to Wazuh when connected
http {
    url => "http://<WAZUH_VM_IP>:55000/logs"
    http_method => "post"
    format => "json"
}
```

Replace <WAZUH\_VM\_IP> with the **actual IP of the Wazuh Ubuntu VM**.

---

### 4. Create a Local Storage Directory

Persistent queues use C:\logstash\data\queue, but also create a **secondary backup** directory:

```
powershell
CopyEdit
mkdir C:\logstash_queue
icacls C:\logstash_queue /grant Everyone:F
```

This ensures logs are written to disk even if Logstash crashes.

---

## 5. Start Logstash

To test the setup:

```
powershell
CopyEdit
cd C:\logstash\bin
.\logstash -f C:\logstash\config\logstash.conf
```

To run Logstash as a **Windows service**, install NSSM (Non-Sucking Service Manager) and run:

```
powershell
CopyEdit
nssm install logstash C:\logstash\bin\logstash.bat -f
C:\logstash\config\logstash.conf
```

---

## ◆ Step 2: Install and Configure Wazuh on Ubuntu VM

Wazuh will receive logs from Logstash once the connection is restored.

### 1. Install Wazuh Server

Run the following on **Ubuntu**:

```
bash
CopyEdit
curl -sO https://packages.wazuh.com/4.4/wazuh-install.sh
sudo bash wazuh-install.sh --wazuh-server --wazuh-indexer --wazuh-dashboard
```

Once installed, open Wazuh Dashboard at:

```
cpt
CopyEdit
http://<WAZUH_VM_IP>:5601
```

Default credentials:

- **Username:** admin

- **Password:** SecretPassword
- 

## 2. Install Filebeat on Ubuntu VM

Filebeat pulls logs from Logstash and forwards them to Wazuh.

```
bash
CopyEdit
curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-
7.10.2-amd64.deb
sudo dpkg -i filebeat-7.10.2-amd64.deb
```

---

## 3. Configure Filebeat to Receive Logs from Logstash

Edit the **Filebeat configuration**:

```
bash
CopyEdit
sudo nano /etc/filebeat/filebeat.yml
```

Add:

```
yaml
CopyEdit
filebeat.inputs:
  - type: filestream
    paths:
      - "/var/log/wazuh/*.log"

output.elasticsearch:
  hosts: ["localhost:9200"]
  index: "wazuh-logs-%{+yyyy.MM.dd}"
```

Save and restart Filebeat:

```
bash
CopyEdit
sudo systemctl restart filebeat
```

---

## ◆ Step 3: Test Log Forwarding

### 1. Verify Logstash Stores Logs When Disconnected

1. **Disconnect Windows from the network.**
2. Generate logs:

```
powershell
CopyEdit
echo "Test log - $(Get-Date)" >> C:\logs\test.log
```

3. Check if logs are stored:

```
powershell  
CopyEdit  
dir C:\logstash\data\queue
```

## 2. Verify Logs Are Forwarded When Reconnected

1. **Reconnect Windows VM to network.**
2. Logstash should start sending logs to Wazuh automatically.
3. Check logs in Wazuh:

```
bash  
CopyEdit  
curl -X GET "localhost:9200/_cat/indices?v"
```

4. Open Wazuh Dashboard → **Discover** → Check for `wazuh-logs-*`.

---

## ◆ Summary

Component	Role
<b>Logstash (Windows 11)</b>	Stores logs <b>on disk</b> if disconnected
<b>Persistent Queue (Windows 11)</b>	Saves logs in <code>C:\logstash\data\queue</code>
<b>Filebeat (Ubuntu VM)</b>	Reads logs from Logstash and sends to Wazuh
<b>Wazuh (Ubuntu VM)</b>	Receives logs, stores in Elasticsearch

- Logs persist even after weeks/months of disconnection.
- Once Windows reconnects, logs are automatically forwarded.

---

## 🚀 Next Steps

Would you like **log rotation**, **alerting**, or **security hardening**? 🚀

---

## How Does Wazuh Start Ingesting Logs in This Setup?

With the **Logstash (Windows) → Wazuh (Ubuntu)** setup, Wazuh ingests logs via **Filebeat**, which pulls logs from Logstash and sends them to Wazuh's indexer. Here's how the logs flow into Wazuh:

### 1. Logstash on Windows

- Reads logs from `C:/logs/*.log`
- Stores logs in **Persistent Queue (C:\logstash\data\queue)** if disconnected
- Forwards logs to Wazuh's **REST API** (`http://<WAZUH_VM_IP>:55000/logs`) when connected

2. **Filebeat on Wazuh (Ubuntu)**
    - o Collects logs from Logstash's HTTP output
    - o Sends logs to **Elasticsearch (Wazuh Indexer)**
  3. **Wazuh Manager**
    - o Reads logs from **Elasticsearch**
    - o Parses logs and applies rules for security alerts
    - o Logs appear in the **Wazuh Dashboard** ([http://<WAZUH\\_VM\\_IP>:5601](http://<WAZUH_VM_IP>:5601))
- 

## ◆ Step-by-Step Configuration to Ingest Logs into Wazuh

Now, let's properly configure **Filebeat** and **Wazuh** to ingest these logs.

---

### ◆ Step 1: Install and Configure Filebeat on Wazuh (Ubuntu VM)

#### 1. Install Filebeat

On **Ubuntu VM (Wazuh Server)**:

```
bash
CopyEdit
curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-
7.10.2-amd64.deb
sudo dpkg -i filebeat-7.10.2-amd64.deb
```

#### 2. Enable Filebeat Wazuh Module

```
bash
CopyEdit
sudo filebeat modules enable wazuh
```

#### 3. Configure Filebeat to Receive Logs from Logstash

Edit **Filebeat configuration**:

```
bash
CopyEdit
sudo nano /etc/filebeat/filebeat.yml
```

Modify:

```
yaml
CopyEdit
filebeat.inputs:
  - type: http_endpoint
    enabled: true
    listen_address: 0.0.0.0
```

```
listen_port: 5044
url: "http://<WAZUH_VM_IP>:55000/logs"
content_type: "application/json"

output.elasticsearch:
  hosts: ["localhost:9200"]
  index: "wazuh-logs-%{+yyyy.MM.dd}"
```

Replace <WAZUH\_VM\_IP> with the actual **Ubuntu VM IP**.

## 4. Restart Filebeat

```
bash
CopyEdit
sudo systemctl restart filebeat
sudo systemctl enable filebeat
```

---

## ◆ Step 2: Configure Wazuh to Process Incoming Logs

### 1. Edit Wazuh Manager Configuration

```
bash
CopyEdit
sudo nano /var/ossec/etc/ossec.conf
```

Add the following **before the <ruleset> section**:

```
xml
CopyEdit
<localfile>
  <log_format>json</log_format>
  <location>/var/log/wazuh/wazuh-http.log</location>
</localfile>
```

Save and restart Wazuh:

```
bash
CopyEdit
sudo systemctl restart wazuh-manager
```

---

## ◆ Step 3: Verify Logs in Wazuh

### 1. Check if Filebeat is Receiving Logs

```
bash
CopyEdit
sudo filebeat test output
```

You should see output confirming that logs are being sent to **Elasticsearch**.

## 2. Check if Wazuh is Parsing Logs

```
bash
CopyEdit
sudo tail -f /var/log/wazuh/wazuh.log
```

Look for logs mentioning **received logs from Filebeat**.

## 3. Verify Logs in Wazuh Dashboard

1. Open **Wazuh Dashboard** ([http://<WAZUH\\_VM\\_IP>:5601](http://<WAZUH_VM_IP>:5601)).
  2. Go to **Discover**.
  3. Select the **wazuh-logs-\*** index pattern.
  4. Check for logs coming from Logstash.
- 

## ◆ Summary

Component	Role
<b>Logstash (Windows 11)</b>	Stores logs <b>on disk</b> when disconnected, forwards logs via HTTP to Wazuh
<b>Filebeat (Ubuntu VM)</b>	Reads logs from Logstash's HTTP output and sends to <b>Elasticsearch</b>
<b>Wazuh Manager</b>	Ingests logs from Elasticsearch, applies security rules
<b>Wazuh Dashboard</b>	Displays logs and alerts

- Logs persist in Logstash on Windows for weeks/months if disconnected
  - Once reconnected, all stored logs are forwarded to Wazuh
  - Logs are indexed in Elasticsearch and visible in Wazuh Dashboard
- 

## 🚀 Next Steps

Would you like:

- **Alerts and rules to detect security threats?**
- **Log filtering to optimize storage?**
- **Hardening for security?**

Let me know how I can refine the setup! 🚀

---

**GENERATING an API KEY**

**Logstash Authentication for Wazuh**

By default, **Wazuh's API requires authentication** to accept logs from Logstash. You need to use a **valid Wazuh API user** in Logstash's HTTP output.

---

## ◆ Step 1: Get or Create Wazuh API Credentials

### 1. Check Existing Wazuh API Users

On your **Ubuntu (Wazuh) VM**, run:

```
bash
CopyEdit
curl -k -X GET "https://localhost:55000/security/users" -H "Authorization: Bearer $(sudo cat /var/ossec/api/configuration/auth/token)" -H "Content-Type: application/json"
```

This will list existing Wazuh API users.

### 2. Create a New API User for Logstash (If Needed)

If you don't have a dedicated user for Logstash, create one:

```
bash
CopyEdit
curl -k -X POST "https://localhost:55000/security/users" -H "Authorization: Bearer $(sudo cat /var/ossec/api/configuration/auth/token)" -H "Content-Type: application/json" -d '{
  "username": "logstash",
  "password": "SecureLogstashPass123",
  "roles": ["administrator"]
}'
```

 Replace `SecureLogstashPass123` with a strong password.

---

## ◆ Step 2: Add Authentication to Logstash Configuration

Edit Logstash's **pipeline file** (`C:\logstash\config\logstash.conf`) on **Windows**:

```
powershell
CopyEdit
notepad C:\logstash\config\logstash.conf
```

Modify the **HTTP output** section to include authentication:

```
yaml
CopyEdit
output {
  http {
    url => "https://<WAZUH_VM_IP>:55000/logs"
```

```

    http_method => "post"
    format => "json"
    headers => {
      "Authorization" => "Basic %{[@metadata][auth_header]}"
      "Content-Type" => "application/json"
    }
  }

filter {
  mutate {
    add_field => { "[@metadata][auth_header]" =>
      "%{logstash_user}:%{logstash_pass}" }
    replace => { "[@metadata][auth_header]" =>
      "${[@metadata][auth_header]}" }
  }
}

```

---

## ◆ Step 3: Securely Store Credentials in Logstash

Instead of hardcoding credentials, store them in **logstash.yml**:

```
powershell
CopyEdit
notepad C:\logstash\config\logstash.yml
```

Add:

```
yaml
CopyEdit
var.logstash_user: "logstash"
var.logstash_pass: "SecureLogstashPass123"
```

Replace **SecureLogstashPass123** with the **actual password** you set.

---

## ◆ Step 4: Restart Logstash

Restart the **Logstash service** to apply the changes:

```
powershell
CopyEdit
Restart-Service logstash
```

Or, if running manually:

```
powershell
CopyEdit
cd C:\logstash\bin
.\logstash -f C:\logstash\config\logstash.conf
```

---

## ◆ Step 5: Verify Authentication is Working

### 1. Check if Logstash Successfully Sends Logs

On **Ubuntu (Wazuh VM)**, check the Wazuh logs:

```
bash
CopyEdit
sudo tail -f /var/ossec/logs/api.log
```

Look for: Successful authentication requests from Logstash

Errors indicating incorrect credentials

### 2. Test API Manually with Credentials

From the **Windows VM**, test authentication using PowerShell:

```
powershell
CopyEdit
$cred =
[Convert]::ToString([Text.Encoding]::UTF8.GetBytes("logstash:SecureLo
gstashPass123"))
Invoke-WebRequest -Uri "https://<WAZUH_VM_IP>:55000/security/users" -
Headers @{Authorization=("Basic " + $cred)} -SkipCertificateCheck
```

If successful, you will see a list of users.

If it fails, check for incorrect passwords or API settings.

---

## ◆ Summary

Component	Configuration
<b>Wazuh API User</b>	Created logstash user with password
<b>Logstash HTTP Output</b>	Added authentication using Wazuh credentials
<b>Stored Credentials Securely</b>	Used logstash.yml to store credentials

- Logstash now securely forwards logs to Wazuh with authentication.  
 Logs are stored if disconnected and sent once reconnected.
- 

## Next Steps

Would you like **API token-based authentication** instead of **Basic Auth** for more security?



## Complete Logstash Config

Here is your **complete Logstash configuration file** (`logstash.conf`) with **Persistent Queue (PQ)** enabled, **authentication for Wazuh API**, and proper **handling of logs** when disconnected.

---

### 📌 Complete Logstash Configuration (Windows 11 VM)

**Location:** `C:\logstash\config\logstash.conf`

```
yaml
CopyEdit
# ----- INPUT: Read logs from files -----
input {
    file {
        path => "C:/logs/*.log"      # Adjust path as needed
        start_position => "beginning"
        since_db_path => "NUL"       # Prevents duplicate log reading after
    restart
    }
}

# ----- FILTER: Add metadata to logs -----
filter {
    mutate {
        add_field => { "[@metadata][auth_header]" =>
"%{logstash_user}: %{logstash_pass}" }
        replace => { "[@metadata][auth_header]" =>
"${[@metadata][auth_header]}" }
    }
}

# ----- OUTPUT: Store & Forward Logs -----
output {
    # Store logs on disk when disconnected (Persistent Backup)
    file {
        path => "C:/logstash_queue/logstash_backup.log"
        codec => "json"
    }

    # Forward logs to Wazuh API when online
    http {
        url => "https://<WAZUH_VM_IP>:55000/logs"
        http_method => "post"
        format => "json"
        headers => {
            "Authorization" => "Basic %{[@metadata][auth_header]}"
            "Content-Type" => "application/json"
        }
    }
}
```

💡 Replace `<WAZUH_VM_IP>` with the actual IP of your Wazuh Ubuntu VM.

---

## ◆ Configure Secure Credential Storage

Instead of storing credentials directly in `logstash.conf`, store them in `logstash.yml`.

**Location:** `C:\logstash\config\logstash.yml`

```
yaml
CopyEdit
var.logstash_user: "logstash"
var.logstash_pass: "SecureLogstashPass123"
```

⚠ Replace `SecureLogstashPass123` with your actual Wazuh API password.

⚠ Ensure `logstash.yml` is **not world-readable** to protect credentials.

---

## ◆ Configure Persistent Queue (Disk Storage)

Logstash stores logs on disk if Wazuh is unreachable.

**Location:** `C:\logstash\config\logstash.yml`

Add these settings:

```
yaml
CopyEdit
# Enable Persistent Queue to store logs on disk
queue.type: persisted
queue.max_bytes: 10gb      # Store up to 10GB of logs before overwriting
queue.checkpoint.acks: 1
queue.checkpoint.writes: 1
queue.drain: true          # Ensures logs are forwarded once Wazuh is back
online

# Define where Logstash stores persistent logs
path.data: C:/logstash/data
```

---

## ◆ Create Log Backup Storage

Create a folder for storing logs **locally** if the connection is lost:

```
powershell
CopyEdit
mkdir C:\logstash_queue
icacls C:\logstash_queue /grant Everyone:F
```

This ensures **logs are always saved on disk** even if Logstash crashes.

---

## ◆ Start Logstash

To start Logstash manually:

```
powershell  
CopyEdit  
cd C:\logstash\bin  
.\\logstash -f C:\\logstash\\config\\logstash.conf
```

To run Logstash as a service (recommended):

```
powershell  
CopyEdit  
nssm install logstash C:\\logstash\\bin\\logstash.bat -f  
C:\\logstash\\config\\logstash.conf
```

Start the service:

```
powershell  
CopyEdit  
Start-Service logstash
```

---

## ◆ Verify Logstash is Working

### 1. Check if Persistent Queue is Active

```
powershell  
CopyEdit  
dir C:\\logstash\\data\\queue
```

 You should see .page files indicating logs are being stored.

### 2. Test API Authentication from Windows

Run the following in PowerShell:

```
powershell  
CopyEdit  
$cred =  
[Convert]::ToString([Text.Encoding]::UTF8.GetBytes("logstash:SecureLo  
gstashPass123"))  
Invoke-WebRequest -Uri "https://<WAZUH_VM_IP>:55000/security/users" -  
Headers @{Authorization=("Basic " + $cred)} -SkipCertificateCheck
```

 If successful, Wazuh API will return a list of users.

### 3. Check Logs in Wazuh

On **Ubuntu (Wazuh VM)**, check the Wazuh logs:

```
bash
CopyEdit
sudo tail -f /var/log/wazuh/wazuh.log
```

- ✓ You should see logs being received from Logstash.

## 4. Verify in Wazuh Dashboard

1. Open **Wazuh Dashboard** ([http://<WAZUH\\_VM\\_IP>:5601](http://<WAZUH_VM_IP>:5601)).
2. Go to **Discover**.
3. Select *wazuh-logs- index\**.
4. ✓ You should see logs coming from Logstash.

---

## ◆ Summary

Component	Role
<b>Logstash (Windows 11)</b>	Reads logs from C:/logs, stores them <b>on disk</b> , and forwards to Wazuh when online.
<b>Persistent Queue (Windows 11)</b>	Stores logs in C:\logstash\data\queue if disconnected from Wazuh.
<b>Filebeat (Ubuntu Wazuh VM)</b>	Reads logs from Logstash and sends them to Wazuh.
<b>Wazuh Manager</b>	Ingests logs from Filebeat and applies security rules.
<b>Wazuh Dashboard</b>	Displays logs and alerts.

- ✓ No logs lost if disconnected for weeks/months
- ✓ Logs are automatically forwarded to Wazuh once reconnected
- ✓ Persistent Queue ensures logs are stored safely on Windows disk

---

## 🚀 Next Steps

Would you like:

- **Log filtering rules?**
- **Alerting on specific log events?**
- **TLS encryption for log forwarding?**

Let me know how I can refine the setup further! 🚀

Gave it to PRO

Below is a **concise, corrected, and verified** guide for setting up **Logstash with a Persistent Queue on Windows 11** to forward logs to **Wazuh on Ubuntu**—with **no data loss** even if disconnected for weeks or months. The instructions also cover **Wazuh installation, API authentication, and Filebeat** (if you choose to route logs via Filebeat).

**Important Note:**

You can either send logs **directly** from Logstash to Wazuh's API or send them **through Filebeat** on the Wazuh side. **Both methods work**, but they are **not both required**. Below, we show **both approaches** so you can pick the one that best fits your needs.

---

# 1. Install and Configure Logstash on Windows

## 1.1 Download and Install Logstash

1. Download [Logstash](#) (ZIP) from Elastic.
2. Extract it to C:\logstash.
3. Create a config folder:

```
powershell  
Copy  
mkdir C:\logstash\config
```

## 1.2 Configure Logstash Persistent Queue

Open C:\logstash\config\logstash.yml in Notepad:

```
yaml  
Copy  
# Enable Persistent Queue (PQ) to store logs on disk  
queue.type: persisted  
queue.max_bytes: 10gb          # Up to 10GB of logs stored on disk  
queue.checkpoint.acks: 1  
queue.checkpoint.writes: 1  
queue.drain: true            # Forward all queued logs once Wazuh is  
back  
  
# Define where Logstash stores queue files  
path.data: C:/logstash/data  
  
# (Optional) Securely store Wazuh API user/pass here:  
var.logstash_user: "logstash"  
var.logstash_pass: "SecureLogstashPass123"
```

**Tip:** Make sure logstash.yml is **not world-readable** (contains credentials).

## 1.3 Create the Logstash Pipeline

Open `C:\logstash\config\logstash.conf` in Notepad:

```
yaml
Copy
# ----- INPUT -----
input {
  file {
    path => "C:/logs/*.log"      # Directory of logs
    start_position => "beginning"
    since_db_path => "NUL"      # Avoid duplicate reads on restart
  }
}

# ----- FILTER -----
filter {
  mutate {
    # If authenticating to Wazuh API, prepare auth header
    add_field => { "[@metadata][auth_header]" =>
"%{logstash_user}:%{logstash_pass}" }
    replace => { "[@metadata][auth_header]" =>
"${[@metadata][auth_header]}" }
  }
}

# ----- OUTPUT -----
output {
  # 1) Store logs locally on disk (backup) in case of disconnection
  file {
    path => "C:/logstash_queue/logstash_backup.log"
    codec => "json"
  }

  # 2) Forward logs to Wazuh once connected
  #     OPTION A: Directly to Wazuh API (bypasses Filebeat)
  http {
    url => "https://<WAZUH_VM_IP>:55000/logs"    # or http:// if you haven't
set up TLS
    http_method => "post"
    format => "json"
    headers => {
      "Authorization" => "Basic %{[@metadata][auth_header]}"
      "Content-Type"  => "application/json"
    }
  }

  #     OPTION B: (Alternative) to an HTTP endpoint on Ubuntu that Filebeat
listens on
  #       IF you want to use Filebeat's http_endpoint input, comment out the
Wazuh API block above
  #       and uncomment the block below:
  #
  # http {
  #   url => "http://<UBUNTU_VM_IP>:5044"
  #   http_method => "post"
  #   format => "json"
  # }
}
```

**Replace** <WAZUH \_ VM \_ IP> with the **Ubuntu VM IP** or hostname.  
Use **https** if you have TLS set up on Wazuh; otherwise, **http** is fine (but less secure).

## 1.4 Create a Local Backup Directory

```
powershell
Copy
mkdir C:\logstash_queue
icacls C:\logstash_queue /grant Everyone:F
```

This ensures logs are also written in plain text locally if needed.

## 1.5 Start Logstash

- **Test run** (in a console window):

```
powershell
Copy
cd C:\logstash\bin
.\logstash -f C:\logstash\config\logstash.conf
```

- **Install as a service** (using NSSM):

```
powershell
Copy
nssm install logstash C:\logstash\bin\logstash.bat -f
C:\logstash\config\logstash.conf
Start-Service logstash
```

---

# 2. Install and Configure Wazuh (Ubuntu)

## 2.1 Install Wazuh Server

```
bash
Copy
curl -sO https://packages.wazuh.com/4.4/wazuh-install.sh
sudo bash wazuh-install.sh --wazuh-server --wazuh-indexer --wazuh-dashboard
```

After installation, access the Wazuh Dashboard at:

```
cpp
Copy
http://<WAZUH _ VM _ IP>:5601
```

Default credentials:

- **Username:** admin
- **Password:** SecretPassword

## 2.2 (Option A) Direct Ingestion from Logstash → Wazuh API

If you **send logs directly** to the Wazuh Manager endpoint (`/logs`), **you do not strictly need Filebeat** for that flow. However, you **must have a Wazuh API user** and **Basic Auth** (or an API Token) for Logstash to post logs.

1. **Create/Check Wazuh API user** (on Ubuntu):

```
bash
Copy
# Check existing users
curl -k -X GET \
  "https://localhost:55000/security/users" \
  -H "Authorization: Bearer $(sudo cat
/var/ossec/api/configuration/auth/token)" \
  -H "Content-Type: application/json"

# Create a user "logstash" if needed
curl -k -X POST \
  "https://localhost:55000/security/users" \
  -H "Authorization: Bearer $(sudo cat
/var/ossec/api/configuration/auth/token)" \
  -H "Content-Type: application/json" \
  -d '{
    "username": "logstash",
    "password": "SecureLogstashPass123",
    "roles": ["administrator"]
}'
```

2. **Update Logstash's `logstash.yml`** with these credentials:

```
yaml
Copy
var.logstash_user: "logstash"
var.logstash_pass: "SecureLogstashPass123"
```

3. **Restart Logstash**, and logs will be sent to the Wazuh API.
4. **Verify** in `/var/ossec/logs/api.log` on Ubuntu:

```
bash
Copy
sudo tail -f /var/ossec/logs/api.log
```

Look for successful postings (HTTP 200).

---

## 2.3 (Option B) Route Logs via Filebeat

Alternatively, if you **want to collect logs using Filebeat** on the Ubuntu VM:

### 2.3.1 Install Filebeat

```
bash
Copy
curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-
7.10.2-amd64.deb
sudo dpkg -i filebeat-7.10.2-amd64.deb
sudo filebeat modules enable wazuh
```

### 2.3.2 Configure Filebeat's `http_endpoint` Input

Edit `/etc/filebeat/filebeat.yml`:

```
yaml
Copy
filebeat.inputs:
  - type: http_endpoint
    enabled: true
    listen_address: 0.0.0.0
    listen_port: 5044
    # Logstash will POST logs here if you configure the Logstash output
with:
    #   http { url => "http://<UBUNTU_VM_IP>:5044" ... }

    # In case you want to store logs as JSON or specify content_type:
    prefix: "/"
    response_code: 200
    response_body: "OK"
    keep_alive: false

output.elasticsearch:
  hosts: ["localhost:9200"]
  index: "wazuh-logs-%{+yyyy.MM.dd}"
```

- **Restart Filebeat:**

```
bash
Copy
sudo systemctl restart filebeat
sudo systemctl enable filebeat
```

- **Modify Logstash `logstash.conf` `output` to:**

```
yaml
Copy
output {
  http {
    url => "http://<UBUNTU_VM_IP>:5044"
    http_method => "post"
    format => "json"
  }
}
```

(Comment out the Wazuh API output block.)

### 2.3.3 Confirm Logs in Wazuh

Because Wazuh Manager and Elasticsearch are on the same VM, logs sent to Elasticsearch (index: `wazuh-logs-*`) will appear in **Wazuh Dashboard → Discover**.

---

## 3. Testing & Verification

1. **Disconnect the Windows VM** (simulate a long outage).
2. Generate logs:

```
powershell  
Copy  
echo "Test log $(Get-Date)" >> C:\logs\test.log
```

3. **Check local queue:**

```
powershell  
Copy  
dir C:\logstash\data\queue
```

You should see .page files storing logs on disk.

4. **Reconnect the Windows VM**; Logstash automatically forwards any queued logs to Wazuh (directly or via Filebeat, depending on which method you chose).
5. **On Ubuntu:**

- o Check **Wazuh logs**:

```
bash  
Copy  
sudo tail -f /var/log/wazuh/wazuh.log
```

- o Check **Elasticsearch indices**:

```
bash  
Copy  
curl -X GET "localhost:9200/_cat/indices?v"
```

Look for wazuh-logs-\*.

- o View in **Wazuh Dashboard** → **Discover** → **wazuh-logs-\*** index.

---

## 4. Summary

Component	Role
<b>Logstash (Windows)</b> 11)	Reads logs from C:/logs, <b>stores them on disk</b> if Wazuh is unreachable, forwards to Wazuh once connected.
<b>Persistent Queue (Windows)</b>	Files in C:\logstash\data\queue ensure <b>no log loss</b> even if offline for weeks/months.
<b>Wazuh Manager (Ubuntu)</b>	Receives logs (either directly from Logstash via API, or via Filebeat) and indexes them in Elasticsearch.

Component	Role
<b>Filebeat (Ubuntu)</b>	(Optional) Intermediary that can receive logs on an HTTP endpoint and forward them into Elasticsearch/Wazuh.
<b>Wazuh Dashboard</b>	Web interface at <code>http://&lt;WAZUH_VM_IP&gt;:5601</code> to search, view, and alert on logs.

## Key Points

- With **Persistent Queue** enabled, Logstash stores logs **on disk** (`C:\logstash\data\queue`).
  - No logs are lost** if the Windows VM is disconnected from the network for days/weeks/months.
  - Once the Windows VM **reconnects**, Logstash **flushes** all stored logs to Wazuh.
  - You can choose **direct ingestion** into Wazuh's API or route logs **through Filebeat** on Ubuntu.
- 

## 5. Next Steps

- Secure the connection** (HTTPS/TLS) for Logstash → Wazuh or Logstash → Filebeat.
- Filter logs** in Logstash (e.g., drop noise, parse fields).
- Configure Wazuh alerts** for specific events or anomalies.
- Implement log rotation** on Windows for `C:\logs\*.log` and for `C:\logstash_queue\logstash_backup.log`.

With this setup, your Windows VM's Logstash will **reliably store and forward** logs to Wazuh, ensuring **no log loss** even during extended outages.

---

### Pro with some context

Below is a **concise and verified** end-to-end setup based on your entire conversation. It ensures **Logstash on Windows** persists logs **on disk** if disconnected from the **Wazuh (Ubuntu)** server for any duration (hours to months) and then automatically forwards them once reconnected, with **Wazuh** ingesting the logs.

---

## 1. High-Level Overview

- Logstash on Windows**
  - Uses a **Persistent Queue** to store logs on disk (`C:\logstash\data\queue`) if the Wazuh server is unreachable.
  - Once the connection is restored, Logstash forwards all queued logs.
- Wazuh on Ubuntu**

- Receives logs either **directly via the Wazuh API** (basic authentication) **or** via **Filebeat** if you prefer an intermediary.
- Logs are stored in **Elasticsearch** (part of the Wazuh stack) and displayed in the **Wazuh Dashboard**.

**Key Point:** You do *not* need Redis or Kafka for local persistence in this scenario. Logstash's built-in **Persistent Queue** fulfills the requirement to store logs on the **Windows VM** itself.

---

## 2. Configure Logstash on Windows

### 2.1 Install Logstash

1. Download the official **Logstash ZIP** from [Elastic](#).
2. Extract it to C:\logstash.
3. Create a config directory:

```
powershell
Copy
mkdir C:\logstash\config
```

### 2.2 Enable Persistent Queue (on-disk storage)

Open C:\logstash\config\logstash.yml and add:

```
yaml
Copy
# Enable persistent queue to store logs on disk
queue.type: persisted
queue.max_bytes: 10gb          # Space for stored logs
queue.checkpoint.acks: 1
queue.checkpoint.writes: 1
queue.drain: true             # Flush queue once reconnected

# Data path for queue .page files
path.data: C:/logstash/data

# (Optional) Define Wazuh credentials here if using Basic Auth
var.logstash_user: "logstash"
var.logstash_pass: "SecureLogstashPass123"
```

**Ensure** logstash.yml isn't world-readable if you store credentials there.

### 2.3 Create the Logstash Pipeline

Open C:\logstash\config\logstash.conf:

```
yaml
Copy
# ----- INPUT -----
```

```

input {
  file {
    path => "C:/logs/*.log"
    start_position => "beginning"
    since_db_path => "NUL"
  }
}

# ----- FILTER -----
filter {
  mutate {
    # Build Basic Auth header if you're posting directly to the Wazuh API
    add_field => {
      "[@metadata][auth_header]" => "%{logstash_user}:#{logstash_pass}"
    }
    replace => {
      "[@metadata][auth_header]" => "${[@metadata][auth_header]}"
    }
  }
}

# ----- OUTPUT -----
output {
  # 1) Store logs locally on disk regardless of connection status
  file {
    path => "C:/logstash_queue/logstash_backup.log"
    codec => "json"
  }

  # 2) Forward logs to Wazuh API when connected
  #     (Comment out if you prefer the Filebeat http_endpoint approach)
  http {
    url => "https://<WAZUH_VM_IP>:55000/logs"
    http_method => "post"
    format => "json"
    headers => {
      "Authorization" => "Basic ${[@metadata][auth_header]}"
      "Content-Type" => "application/json"
    }
  }
}

# Alternatively, if sending logs to Filebeat's http_endpoint on Ubuntu:
# http {
#   url => "http://<UBUNTU_VM_IP>:5044"
#   http_method => "post"
#   format => "json"
# }
}

```

**Replace** <WAZUH\_VM\_IP> with your Ubuntu server's IP. Use `http` if TLS is *not* set up on Wazuh, but `https` is recommended for security.

## 2.4 Create a Local Backup Folder

```

powershell
Copy
mkdir C:\logstash_queue
icacls C:\logstash_queue /grant Everyone:F

```

This is just an extra file-based backup location (`logstash_backup.log`). The **primary** on-disk storage mechanism is still the **Persistent Queue** in `C:\logstash\data\queue`.

## 2.5 Start Logstash

- **Test run** in a PowerShell window:

```
powershell
Copy
cd C:\logstash\bin
.\logstash -f C:\logstash\config\logstash.conf
```

- **Install as a Windows service** (using NSSM):

```
powershell
Copy
nssm install logstash C:\logstash\bin\logstash.bat -f
C:\logstash\config\logstash.conf
Start-Service logstash
```

---

## 3. Set Up Wazuh on Ubuntu

### 3.1 Install Wazuh (Server + Indexer + Dashboard)

```
bash
Copy
curl -sO https://packages.wazuh.com/4.4/wazuh-install.sh
sudo bash wazuh-install.sh --wazuh-server --wazuh-indexer --wazuh-dashboard
```

After installation, the dashboard is at:

```
cpp
Copy
http://<WAZUH_VM_IP>:5601
```

Default credentials are usually `admin / SecretPassword`.

### 3.2 (Option A) Direct Wazuh API Ingestion

If you're using the `direct http { url => "https://<WAZUH_VM_IP>:55000/logs" }` approach, you'll need:

1. A **Wazuh API user** with appropriate permissions.
2. The correct **username** and **password** in `logstash.yml` or `logstash.conf`.

Create or check an API user on Ubuntu:

```
bash
Copy
# List existing Wazuh API users
```

```

curl -k -X GET "https://localhost:55000/security/users" \
-H "Authorization: Bearer $(sudo cat
/var/ossec/api/configuration/auth/token)"

# Create a new user (if needed)
curl -k -X POST "https://localhost:55000/security/users" \
-H "Authorization: Bearer $(sudo cat
/var/ossec/api/configuration/auth/token)" \
-H "Content-Type: application/json" \
-d '{
    "username": "logstash",
    "password": "SecureLogstashPass123",
    "roles": ["administrator"]
}'

```

**Match** the username/password with what you put in **Logstash**.

### 3.3 (Option B) Use Filebeat as an Intermediary

If you'd rather have **Logstash** → **Filebeat** → Wazuh:

1. **Install Filebeat** on Ubuntu:

```

bash
Copy
curl -L -O
https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-
7.10.2-amd64.deb
sudo dpkg -i filebeat-7.10.2-amd64.deb

```

2. **Configure** an `http_endpoint` in `/etc/filebeat/filebeat.yml`:

```

yaml
Copy
filebeat.inputs:
  - type: http_endpoint
    enabled: true
    listen_address: 0.0.0.0
    listen_port: 5044
    prefix: "/"
    response_code: 200
    response_body: "OK"

output.elasticsearch:
  hosts: ["localhost:9200"]
  index: "wazuh-logs-%{+yyyy.MM.dd}"

```

3. **Modify** Logstash's `http` output to:

```

yaml
Copy
output {
  http {
    url => "http://<UBUNTU_VM_IP>:5044"
    http_method => "post"
    format => "json"
  }
}

```

```
}
```

4. **Restart Filebeat** and check logs in the Wazuh dashboard (Elasticsearch).
- 

## 4. Testing & Validation

1. **Disconnect Windows VM** from the network to simulate a multi-week outage.
2. Generate a new test log:

```
powershell
Copy
echo "Offline Log - $(Get-Date)" >> C:\logs\test.log
```

3. Check **Persistent Queue** storage:

```
powershell
Copy
dir C:\logstash\data\queue
```

You'll see .page files indicating logs are being saved on disk.

4. **Reconnect the Windows VM**.

- o Logstash flushes all .page files to Wazuh automatically.
- o On Ubuntu, verify logs are received:

```
bash
Copy
# If direct to Wazuh API
sudo tail -f /var/ossec/logs/api.log

# If via Filebeat
sudo filebeat test output
```

- o In the **Wazuh Dashboard** ([http://<WAZUH\\_VM\\_IP>:5601](http://<WAZUH_VM_IP>:5601)), go to **Discover**, choose the `wazuh-logs-*` index, and confirm logs appear.
- 

## 5. Summary

This setup is correct given the conversation and your requirements:

- **Logstash on Windows** enables **Persistent Queue** and writes .page files to `C:\logstash\data\queue`.
- If the Windows VM cannot reach Wazuh for days/months, the logs **accumulate on disk**.
- Once the connection is restored, all **stored logs** are forwarded.

- **Wazuh** receives logs either directly via the **Wazuh API** or **Filebeat** (depending on which method you choose).

<b>Component</b>	<b>Role</b>
<b>Logstash (Windows 11)</b>	Reads logs from C:\logs, stores them on disk if offline, and forwards them upon reconnection.
<b>Persistent Queue (Windows)</b>	Resides in C:\logstash\data\queue for on-disk resilience.
<b>Wazuh Manager (Ubuntu)</b>	Ingests logs (via API or via Filebeat → Elasticsearch).
<b>Wazuh Dashboard</b>	Displays the logs and allows searching, alerting, etc.

Thus, **no logs are lost** even if the connection is down for an extended period. Once reconnected, **all queued logs** flow into Wazuh for analysis and visibility in the dashboard.

**Yes**—the final instructions in your conversation are **correct** for achieving resilient local (Windows-based) log storage and forwarding to Wazuh.

o1-pro

Search  
Deep research  
ChatGPT can mak