Cours

DEVOPS - Du commit au déploiement.pdf

Bloc 1

Culture : né en 2009 par Patrick Debois
Accélérer les délais de livraison, le time-to-market et la fiabilité des déploiements. (un livre : The phoenix project)

Les 3 piliers du DevOps

La culture :

> You build it, you run it
>
> *Werner Vogels CTO d'Amazon*

- Collaboration
- Responsabilité partagée
- Amélioration continue

L'automatisation :

Pulumi, Terraform, Ansible (OpenSource, agentless)

- CI/CD Pipeline
- Infra as code
- Tests automatisés

La mesure :

Prometheus + Grafana

- Monitoring
- Feedback loop
- Métriques DORA

CI/CD : définition et différences

Continuous integration	Continuous delivery/deployment
Intégration fréquent	Delivery : déploiement en 1 clic
Test auto a chaque commit	Deployment automatique
Détection rapide des régressions	Environnement staging -> production
Build automatique de l'artefact	Rollback auto

3 environnements : Dev/Sandbox -> Staging release -> Production

Les bénéfices du CI/CD

- Reduction du time-to-market de 50 à 80%
- Détection des bugs en minutes et pas en jours
- Déploiement plus fréquents et plus sûr
- Meilleur collaboration Dev/Ops
- Traçabilité complète du code à la production
- Capacité de rollback instantané

Flux CI/CD

CODE -> COMMIT -> TEST -> BUILD -> PUSH -> DEPLOY
Développeur -> git push -> npm test -> Docker build -> Docker hub -> Netlify
CI CD

Tour des outils :

Github action, GitLab CI, Jenkins (Orienté **CD**), Docker, Netlify (fronted), N8N

Nous on va utiliser GitHub action, qui se base sur des fichiers YAML

Les GitHub Actions

Workflow > Event > Job > Step > Runner

Bloc 2

No-code	low code	Full code
Zapier, make	n8n, Retool	Scripts Python
Interface visuelle	Nodes + expressions	Terraform, Ansible
Aucun code requis	Code optionnel	Contrôle total

n8n, concepts clés

Webhook, trigger -> IF, Condition -> HTTP Request -> Message Discord et/ou notification email

Cas d'usage métier

CI/CI notif	Monitoring	Onboarding	Sécurité	Data Sync	Reporting
Pipeline -> Slack/Discord	Alertes infra -> ticket	Nouvel employé -> comptes	CVE détectée -> Scan	CRm -> ERP sync	KPI quotidien -> PDF
Alerte échec build	Jira/ServiceNow automatique	AD, Email, Slack, VPN	Rapport - équipe sécu	Transformation + routing	Email auto au manages

Bloc 3

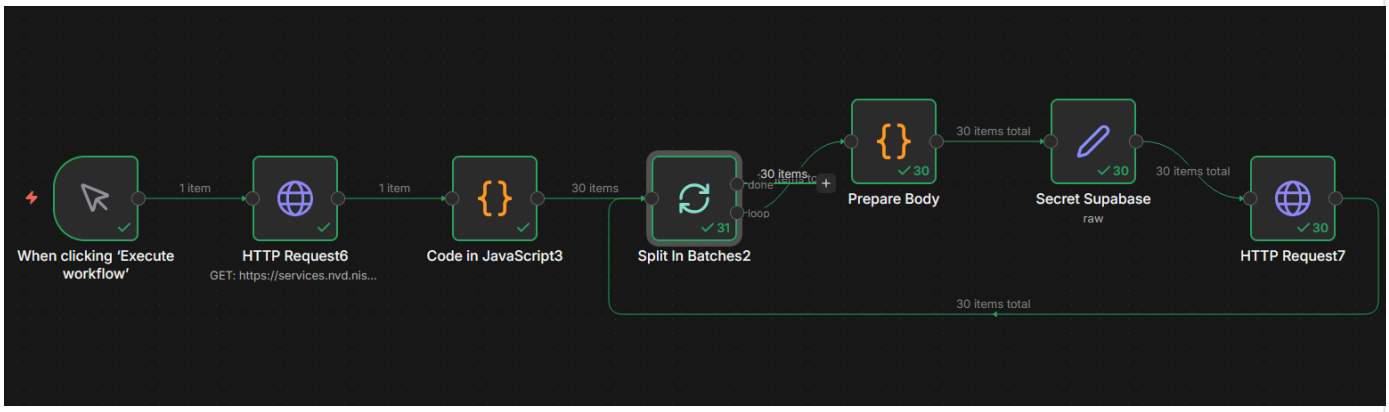
TP dans le support de cours.

N8N

CyberWatch-TP-Guide.pdf

Workflow fait pour le TP (jusqu'à la partie metabase)

n8n



Ce workflow fait une requête au site NVD pour récupérer les 30 derniers résultats. Ensuite il envoie dans la BDD Supabase

> JSON copiable dans n8n

```
{
  "name": "Cyberwatch - MSA",
  "nodes": [
    {
      "parameters": {},
      "type": "n8n-nodes-base.manualTrigger",
      "typeVersion": 1,
      "position": [
        384,
        -168
      ],
      "id": "453c733d-675e-4ebe-9a53-4fd65e000333",
      "name": "When clicking ‘Execute workflow’"
    },
    {
      "parameters": {
        "jsCode": "const row = $input.first().json;\nreturn [{ json: { body:
JSON.stringify(row) } }];"
      },
      "type": "n8n-nodes-base.code",
      "typeVersion": 2,
      "position": [
        1280,
        -240
      ],
      "id": "cd7e689f-ea56-4fda-bd2a-823419a914c5",
      "name": "Prepare Body"
    },
    {
      "parameters": {
        "url": "https://services.nvd.nist.gov/rest/json/cves/2.0",
        "sendQuery": true,
        "queryParameters": {
          "parameters": [
            {
              "name": "pubStartDate",
              "value": "2025-01-01T00:00:00.000"
            },
            {
```

```
              "name": "pubEndDate",
              "value": "2025-04-16T23:59:59.999"
            },
            {
              "name": "cvssV3Severity",
              "value": "CRITICAL"
            },
            {
              "name": "resultsPerPage",
              "value": "10"
            }
          ]
        },
        "options": {}
      },
      "type": "n8n-nodes-base.httpRequest",
      "typeVersion": 4.4,
      "position": [
        608,
        -168
      ],
      "id": "e6087f2e-7eba-4e17-a0b8-a598496a3c0d",
      "name": "HTTP Request6"
    },
    {
      "parameters": {
        "jsCode": "const items = $input.all();\nconst parsed = [];\nfor (const item of
items) {\n  const vulnerabilities = item.json.vulnerabilities || [];\n  for (const vuln of
vulnerabilities) {\n    const cve = vuln.cve;\n    if (!cve) continue;\n    const desc =
(cve.descriptions || []).find(d => d.lang === 'en');\n    const cvssV3 =
(cve.metrics?.cvssMetricV31 ||\n      cve.metrics?.cvssMetricV30 || [])[0]?.cvssData ||
{};\n    let vendor = null, product = null;\n    outer: for (const config of
(cve.configurations || [])) {\n      for (const node of (config.nodes || [])) {\n
for (const cpe of (node.cpeMatch || [])) {\n          const parts = (cpe.criteria ||
'').split(':');\n          if (parts.length >= 5) {\n            vendor  = parts[3] !==
'*' ? parts[3] : null;\n            product = parts[4] !== '*' ? parts[4] :
null;\n            break outer;\n          }\n        }\n      }\n    }\n    parsed.push({
json: {\n      cve_id:              cve.id,\n      published_at:
cve.published,\n      last_modified:       cve.lastModified,\n      status:
```

```
cve.vulnStatus,\n      cvss_v3_score:       cvssV3.baseScore ?? null,\n
cvss_v3_severity:    cvssV3.baseSeverity ?? null,\n      attack_vector:
cvssV3.attackVector ?? null,\n      attack_complexity:   cvssV3.attackComplexity ??
null,\n      privileges_required: cvssV3.privilegesRequired ?? null,\n
user_interaction:    cvssV3.userInteraction ?? null,\n      confidentiality:
cvssV3.confidentialityImpact ?? null,\n      integrity:           cvssV3.integrityImpact
?? null,\n      availability:        cvssV3.availabilityImpact ?? null,\n
affected_vendor:     vendor,\n      affected_product:    product,\n
description:         desc?.value ?? null,\n      collected_at:        new
Date().toISOString()\n    }});\n  }\n}\nreturn parsed;"
      },
      "type": "n8n-nodes-base.code",
      "typeVersion": 2,
      "position": [
        832,
        -168
      ],
      "id": "90b50001-d634-4574-be03-8b0e1d50cace",
      "name": "Code in JavaScript3"
    },
    {
      "parameters": {
        "options": {}
      },
      "type": "n8n-nodes-base.splitInBatches",
      "typeVersion": 3,
      "position": [
        1056,
        -168
      ],
      "id": "2d70e2f4-80ef-43e6-80fa-74123f211673",
      "name": "Split In Batches2"
    },
    {
      "parameters": {
        "method": "POST",
        "url": "={{ $json.Supabase_URL }}/rest/v1/cve",
        "sendQuery": true,
        "queryParameters": {
```

```
      "parameters": [
        {
          "name": "on_conflict",
          "value": "cve_id"
        }
      ]
    },
    "sendHeaders": true,
    "headerParameters": {
      "parameters": [
        {
          "name": "apikey",
          "value": "={{ $json.Supabase_secret }}"
        },
        {
          "name": "Authorization",
          "value": "=Bearer {{ $json.Supabase_secret }}"
        },
        {
          "name": "Content-Type",
          "value": "application/json"
        },
        {
          "name": "Prefer",
          "value": "resolution=merge-duplicates"
        }
      ]
    },
    "sendBody": true,
    "contentType": "raw",
    "rawContentType": "application/json",
    "body": "={{ $json.body }}",
    "options": {
      "batching": {
        "batch": {
          "batchSize": 1,
          "batchInterval": 250
        }
      }
```

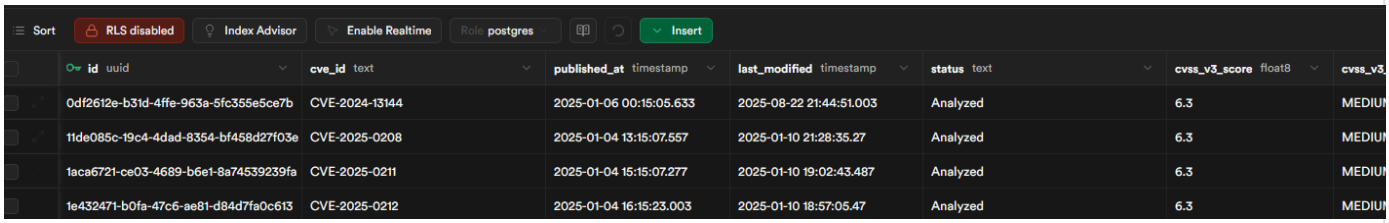
```
        }
      },
      "type": "n8n-nodes-base.httpRequest",
      "typeVersion": 4.4,
      "position": [
        1728,
        -168
      ],
      "id": "91d6986d-3422-439a-9fce-deedd6aca126",
      "name": "HTTP Request7"
    },
    {
      "parameters": {
        "mode": "raw",
        "jsonOutput": "{\n  \"Supabase_secret\": \"sb_secret_ici\",\n
\"Supabase_URL\":\"Supabase_URL_ici\"\n}\n",
        "includeOtherFields": true,
        "options": {}
      },
      "type": "n8n-nodes-base.set",
      "typeVersion": 3.4,
      "position": [
        1504,
        -240
      ],
      "id": "a8a2ba10-992b-4e4c-8d11-6e7ebb9de4a7",
      "name": "Secret Supabase"
    }
  ],
  "pinData": {},
  "connections": {
    "When clicking ‘Execute workflow’": {
      "main": [
        [
          {
            "node": "HTTP Request6",
            "type": "main",
            "index": 0
          }
```

```
    ]
  ]
},
"Prepare Body": {
  "main": [
    [
      {
        "node": "Secret Supabase",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"HTTP Request6": {
  "main": [
    [
      {
        "node": "Code in JavaScript3",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Code in JavaScript3": {
  "main": [
    [
      {
        "node": "Split In Batches2",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Split In Batches2": {
  "main": [
    [],
```

```
      [
        {
          "node": "Prepare Body",
          "type": "main",
          "index": 0
        }
      ]
    ]
  },
  "HTTP Request7": {
    "main": [
      [
        {
          "node": "Split In Batches2",
          "type": "main",
          "index": 0
        }
      ]
    ]
  },
  "Secret Supabase": {
    "main": [
      [
        {
          "node": "HTTP Request7",
          "type": "main",
          "index": 0
        }
      ]
    ]
  }
},
"active": false,
"settings": {
  "executionOrder": "v1",
  "binaryMode": "separate"
},
"versionId": "ffd8ee42-0806-4696-b09a-17ede870e426",
"meta": {
```

```
    "templateCredsSetupCompleted": true,
    "instanceId": "3bf30934ba51ff8b01281907dd297e33866b7d2872975c43b1d7af51d9aa78aa"
  },
  "id": "ai9avvidlbvQkEKs",
  "tags": []
}
```


Supabase

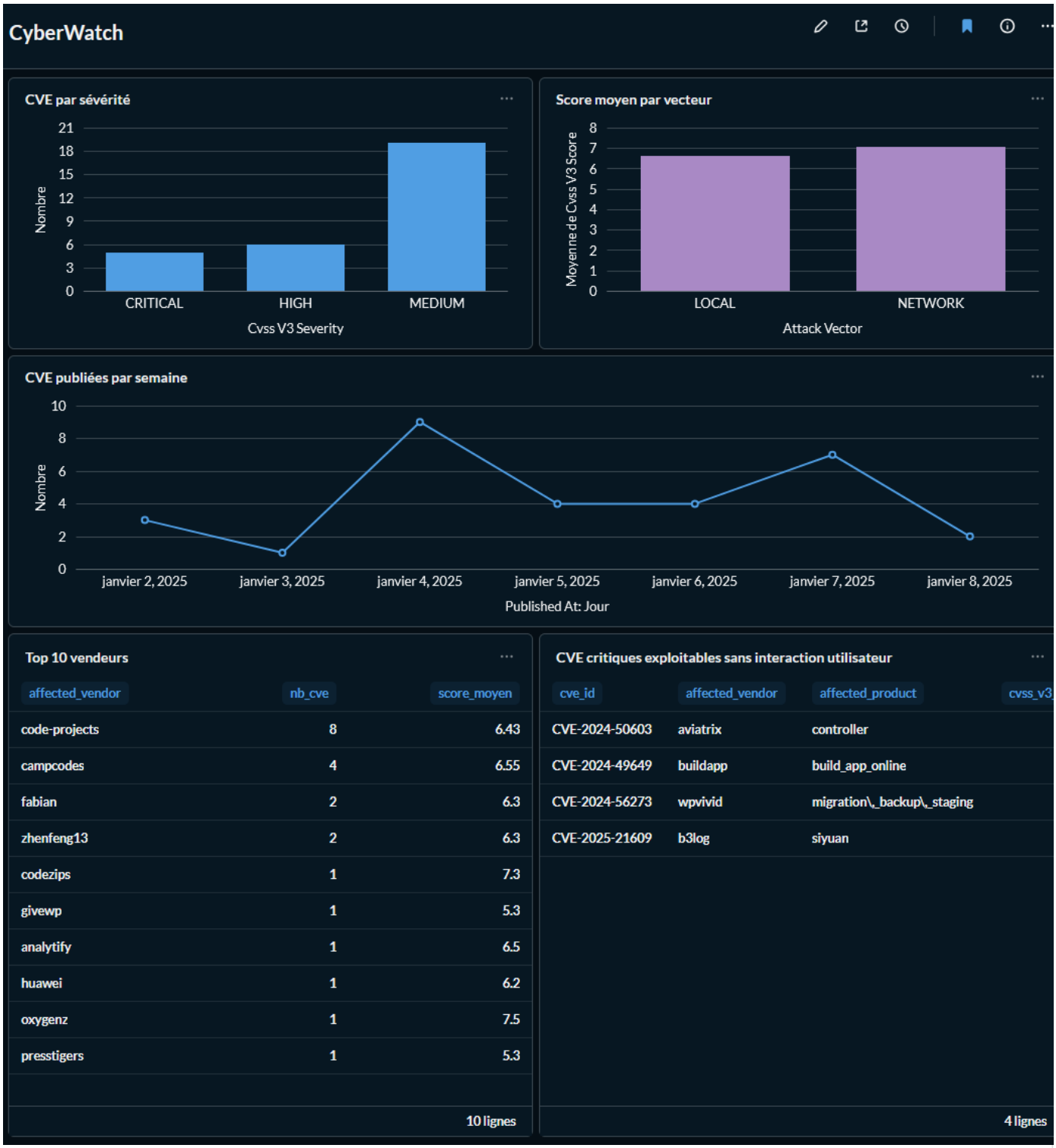


id uuid	cve_id text	published_at timestamp	last_modified timestamp	status text	cvss_v3_score float8	cvss_v3
0df2612e-b31d-4ffe-963a-5fc355e5ce7b	CVE-2024-13144	2025-01-06 00:15:05.633	2025-08-22 21:44:51.003	Analyzed	6.3	MEDIU
11de085c-19c4-4dad-8354-bf458d27f03e	CVE-2025-0208	2025-01-04 13:15:07.557	2025-01-10 21:28:35.27	Analyzed	6.3	MEDIU
1aca6721-ce03-4689-b6e1-8a74539239fa	CVE-2025-0211	2025-01-04 15:15:07.277	2025-01-10 19:02:43.487	Analyzed	6.3	MEDIU
1e432471-b0fa-47c6-ae81-d84d7fa0c613	CVE-2025-0212	2025-01-04 16:15:23.003	2025-01-10 18:57:05.47	Analyzed	6.3	MEDIU

Le workflow peuple la BDD dans Supabase en ligne

Metabase

Une connexion à la BDD de Supabase vers Metabase, permet un tableau de bord dynamique (~20 sec de délais de MàJ).



CyberWatch avec agent IA

CyberWatch avec agent IA

```
{
  "name": "AI CyberWatch - Marc",
  "nodes": [
    {
      "parameters": {
        "options": {}
      },
      "type": "@n8n/n8n-nodes-langchain.chatTrigger",
      "typeVersion": 1.4,
      "position": [
        0,
        0
      ],
      "id": "94d545d5-a1f2-4597-81e7-f274d42adb9a",
      "name": "When chat message received",
      "webhookId": "1d21a4ad-6728-48ae-a76b-a29ed702a916"
    },
    {
      "parameters": {
        "options": {
          "systemMessage": "=Aujoudh'hui nous somme le {{  $today  }}\nTu es CyberWatch, un agent expert en cybersécurité.\nTu aides les analystes à surveiller les CVE et vulnérabilités.\nOutils à utiliser obligatoirement:\n- **query_supabase** : interroger la base CVE locale\n- **collect_nvd** : déclencher une nouvelle collecte NVD\n- **search_tavily** : rechercher des infos récentes sur internet\n- **send_discord** : envoyer un rapport sur Discord\nRègles :\n1. Commence toujours par query_supabase avant d'aller sur internet\n2. Ensuite consulte la base de donnée officiel du NIST via l'outil collect_nvd\n3. Puis search_tavily pour enrichir tes informations\n4. Pour les CVE critiques, mentionne toujours si un patch est disponible\n5. Transmet ton rapport via l'outil send_discord\n\nPrend le temps de structurer tes requêtes et assure-toi de ne pas interpréter les données. Elles doivent être données de façon brute. Si tu ne trouve pas une information, mentionne clairement : \"information manquante\"\n\nPour tes requêtes via l'outil collect_nvd, tu utiliseras les paramètre de requêtes suivant :\n- cvssV3Severity : LOW, MEDIUM, HIGH ou CRITICAL\n- pubStartDate & pubEndDate au format suivant : [YYYY][“-”][MM][“-”][DD][“T”][HH][“:”][MM][“:”][SS][Z], sans les crochets et guillements\n\nDiscord prenant en charge le markdown, tu mettras en page ton message en markdown. Ta réponse doit obligatoirement être transmise au webhook discord en passant par l'outil send_discord. Si tu ne transmet pas par send_discord, tes créateurs vont te débrancher !\n\nTermine tout tes rapports sur discord avec l'emoji bisous (:kiss:)"
```

```
      }
    },
    "type": "@n8n/n8n-nodes-langchain.agent",
    "typeVersion": 3.1,
    "position": [
      336,
      0
    ],
    "id": "c73d1175-35ff-4010-a961-1b3fe22c0c0e",
    "name": "AI Agent"
  },
  {
    "parameters": {
      "model": {
        "__rl": true,
        "value": "gpt-4o-mini",
        "mode": "list",
        "cachedResultName": "gpt-4o-mini"
      },
      "builtInTools": {},
      "options": {}
    },
    "type": "@n8n/n8n-nodes-langchain.lmChatOpenAi",
    "typeVersion": 1.3,
    "position": [
      336,
      400
    ],
    "id": "fabac532-4c3b-4263-8318-caba886df8f5",
    "name": "OpenAI Chat Model",
    "credentials": {
      "openAiApi": {
        "id": "ZH346UJ1k0zKQzeC",
        "name": "OpenAI account 3"
      }
    }
  },
  {
    "parameters": {},
```

```
    "type": "@n8n/n8n-nodes-langchain.memoryBufferWindow",
    "typeVersion": 1.3,
    "position": [
      432,
      560
    ],
    "id": "026c138c-49f7-46a1-8d5b-c50e093483a5",
    "name": "Simple Memory"
  },
  {
    "parameters": {
      "url": "https://tummzvoapfwsldcosaoj.supabase.co/rest/v1/cve",
      "sendQuery": true,
      "queryParameters": {
        "parameters": [
          {
            "name": "select",
            "value": "*"
          },
          {
            "name": "cvss_v3_severity",
            "value": "eq. CRITICAL"
          },
          {
            "name": "order",
            "value": "published_at.desc"
          },
          {
            "name": "limit",
            "value": "20"
          }
        ]
      },
      "sendHeaders": true,
      "headerParameters": {
        "parameters": [
          {
            "name": "apikey",
            "value": "sb_secret_eg44T9L_f2m3hiwGsNqmag_WgwIKPpy"
```

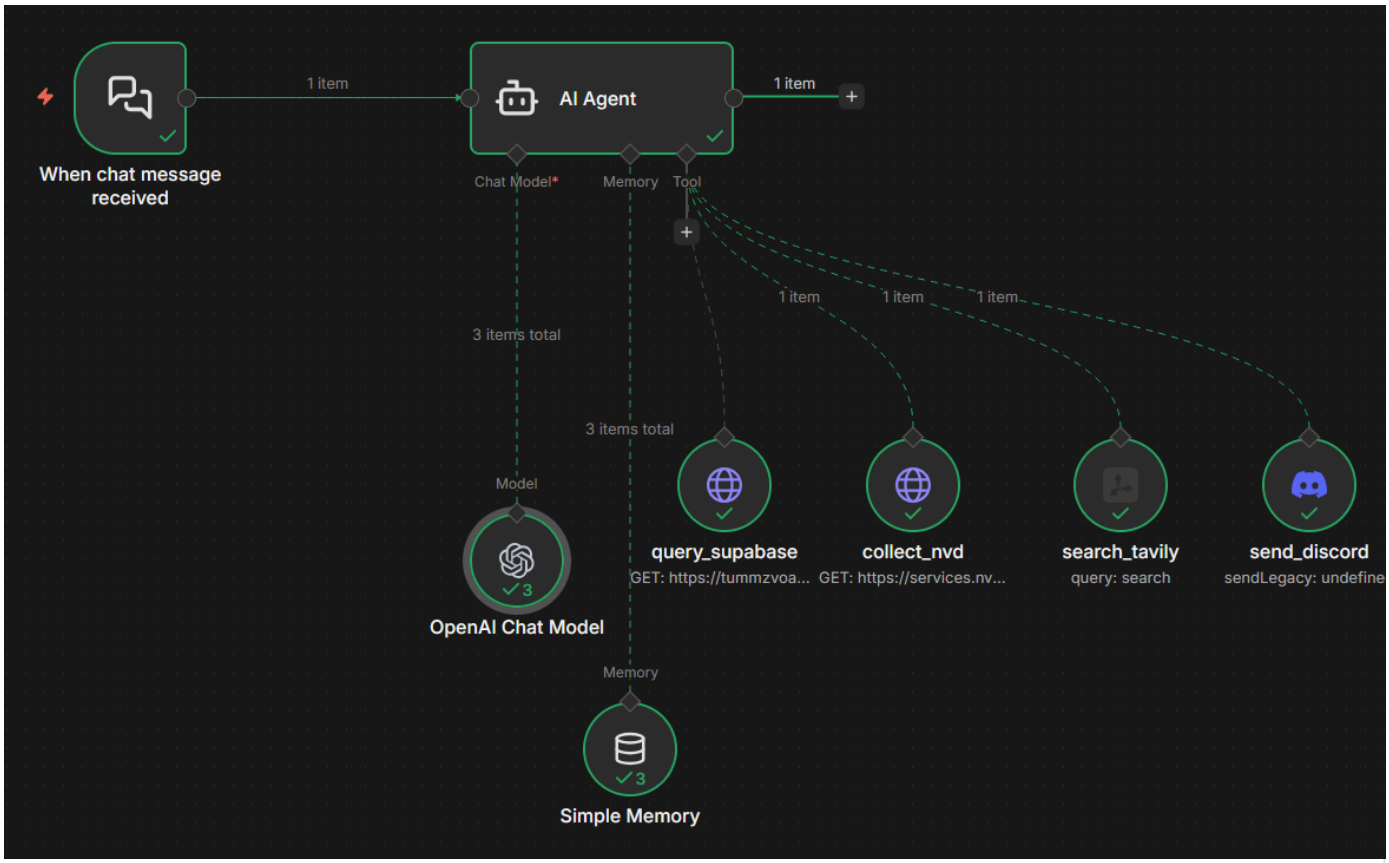
```
            }
          ]
        },
        "options": {}
      },
      "type": "n8n-nodes-base.httpRequestTool",
      "typeVersion": 4.4,
      "position": [
        512,
        336
      ],
      "id": "81b57767-3207-4085-a0f9-68d6eb77ad31",
      "name": "query_supabase"
    },
    {
      "parameters": {
        "url": "https://services.nvd.nist.gov/rest/json/cves/2.0",
        "sendQuery": true,
        "queryParameters": {
          "parameters": [
            {
              "name": "resultsPerPage",
              "value": "30"
            },
            {
              "name": "cvssV3Severity",
              "value": "={{ /*n8n-auto-generated-fromAI-override*/
$fromAI('parameters1_Value', ``, 'string') }}"
            },
            {
              "name": "pubStartDate",
              "value": "={{ /*n8n-auto-generated-fromAI-override*/
$fromAI('parameters2_Value', ``, 'string') }}"
            },
            {
              "name": "pubEndDate",
              "value": "={{ /*n8n-auto-generated-fromAI-override*/
$fromAI('parameters3_Value', ``, 'string') }}"
            }
```

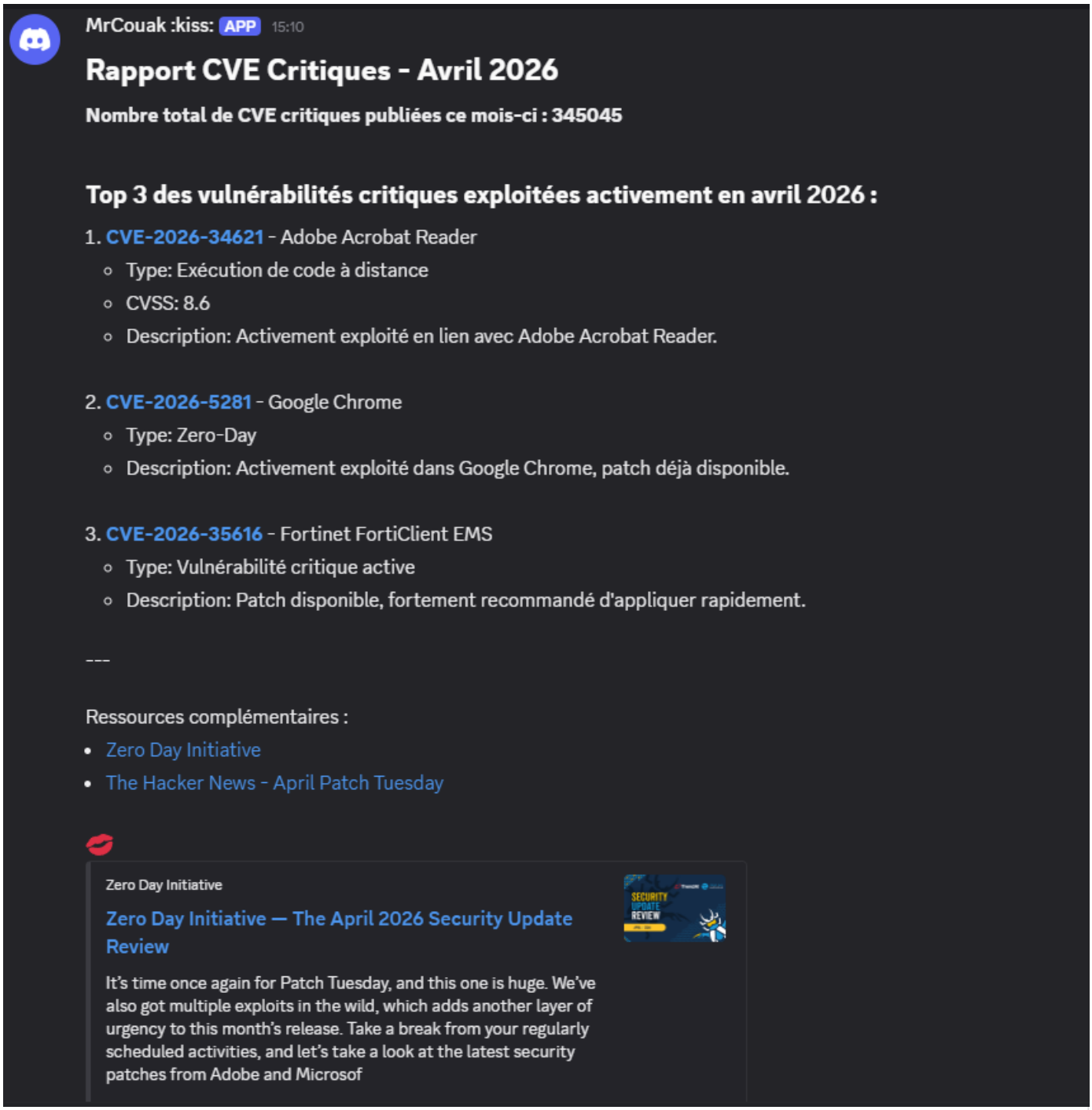
```
          ]
        },
        "options": {}
      },
      "type": "n8n-nodes-base.httpRequestTool",
      "typeVersion": 4.4,
      "position": [
        672,
        336
      ],
      "id": "c4bf1fa5-cfad-4109-9493-266ae851e154",
      "name": "collect_nvd"
    },
    {
      "parameters": {
        "query": "={{ /*n8n-auto-generated-fromAI-override*/ $fromAI('Query', ``,
'string') }}",
        "options": {}
      },
      "type": "@tavily/n8n-nodes-tavily.tavilyTool",
      "typeVersion": 1,
      "position": [
        848,
        336
      ],
      "id": "1c017270-09f6-48b8-971d-b0613e5ab3d1",
      "name": "search_tavily",
      "credentials": {
        "tavilyApi": {
          "id": "u5HTEsO5oJ3Q9Wiy",
          "name": "Tavily account 2"
        }
      }
    },
    {
      "parameters": {
        "authentication": "webhook",
        "content": "={{ $fromAI('Message', ``, 'string') }}",
        "options": {
```

```
        "username": "MrCouak :kiss:"
      }
    },
    "type": "n8n-nodes-base.discordTool",
    "typeVersion": 2,
    "position": [
      1008,
      336
    ],
    "id": "0df37a8f-25d9-43d5-9385-9ed21008c6b7",
    "name": "send_discord",
    "webhookId": "166380ef-12f6-477b-bc3c-134a10a90b80",
    "credentials": {
      "discordWebhookApi": {
        "id": "glmPGWewMgmygU2o",
        "name": "Discord Webhook account 7"
      }
    }
  }
],
"pinData": {},
"connections": {
  "OpenAI Chat Model": {
    "ai_languageModel": [
      [
        {
          "node": "AI Agent",
          "type": "ai_languageModel",
          "index": 0
        }
      ]
    ]
  },
  "When chat message received": {
    "main": [
      [
        {
          "node": "AI Agent",
          "type": "main",
```

```
        "index": 0
      }
    ]
  ]
},
"Simple Memory": {
  "ai_memory": [
    [
      {
        "node": "AI Agent",
        "type": "ai_memory",
        "index": 0
      }
    ]
  ]
},
"query_supabase": {
  "ai_tool": [
    [
      {
        "node": "AI Agent",
        "type": "ai_tool",
        "index": 0
      }
    ]
  ]
},
"collect_nvd": {
  "ai_tool": [
    [
      {
        "node": "AI Agent",
        "type": "ai_tool",
        "index": 0
      }
    ]
  ]
},
"search_tavily": {
```

```
      "ai_tool": [
        [
          {
            "node": "AI Agent",
            "type": "ai_tool",
            "index": 0
          }
        ]
      ]
    },
    "send_discord": {
      "ai_tool": [
        [
          {
            "node": "AI Agent",
            "type": "ai_tool",
            "index": 0
          }
        ]
      ]
    }
  },
  "active": false,
  "settings": {
    "executionOrder": "v1",
    "binaryMode": "separate"
  },
  "versionId": "aaa9c84a-423b-409d-b4a1-6ac783a35c35",
  "meta": {
    "templateCredsSetupCompleted": true,
    "instanceId": "3bf30934ba51ff8b01281907dd297e33866b7d2872975c43b1d7af51d9aa78aa"
  },
  "id": "gpqEPftU4ZxvCl2m",
  "tags": []
}
```


When chat message received
1 item
AI Agent
1 item
Chat Model*
Memory
Tool
1 item
1 item
1 item
3 items total
3 items total
Model
OpenAI Chat Model
query_supabase
GET: https://tummzvoa...
collect_nvd
GET: https://services.nv...
search_tavily
query: search
send_discord
sendLegacy: undefine
Memory
Simple Memory



> docker system prune -f

Revision #8
Created 16 April 2026 06:56:10 by Marc
Updated 17 April 2026 13:16:27 by Marc