GlobalNOC NOCBot

Experimenting with LLMs for Network Operations

What was the goal?

- Investigate different AI technologies to better understand and develop some skills, ideas, tools and future vision for AI at the GlobalNOC
- Experiment with a lot of different software, and services available to us
- Attempt to integrate several different aspects of the GlobalNOC Environment to see what is possible and where we can go with AI
- Mostly - Learn by doing, and have a better understanding of what is and isn't possible today

GlobalNOC
at Indiana University

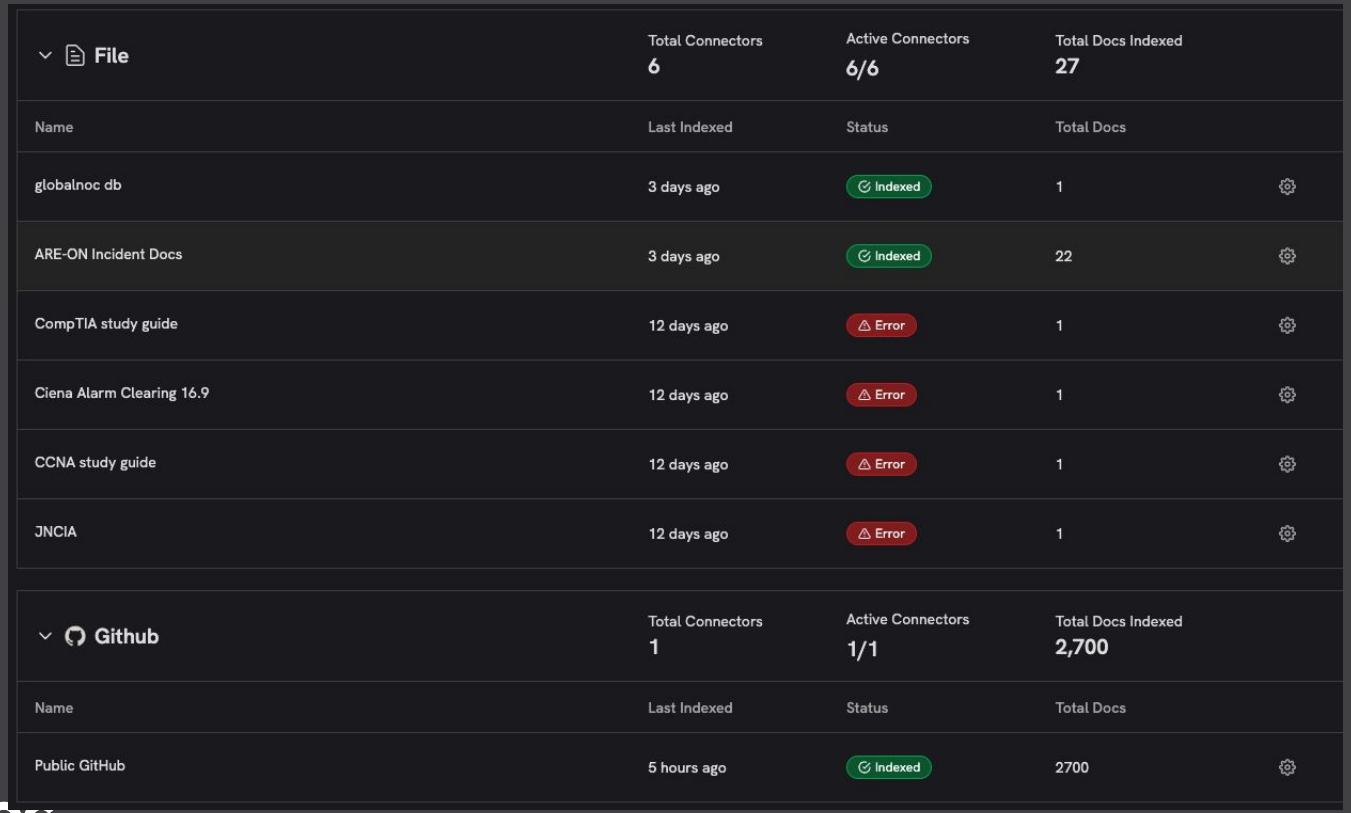
RAG and Software Evaluation

- Experimented with Multiple RAG systems
  - N8N
  - Onyx
  - Ragflow
- Ragflow was quickly eliminated, it was much more limited in the “connectors” (note is has made pretty great strides since i first tried it out in April)
- N8N is good, but requires a bit more configuration as I would have to build my own workflows for a basic RAG to function
- Onyx comes out of the box basically ready to go, with lots of connectors, and the ability to do actions (workflows coming soon)

Onyx - what is it?

- Platform for Generative AI - can work with many different LLMs (Locally hosted or cloud based)
- Built-in slackbot (slack users can chat with it)
- Allows for custom document sets for different users
  - Can create a separate document repository for Network Engineers, Customers, Service Desk and Systems Engineers
- Tool/Action/MCP integration
- Permissions to use custom built agents with different Tools,

Documents



File

Total Connectors	Active Connectors	Total Docs Indexed
6	6/6	27

Name	Last Indexed	Status	Total Docs
globalnoc db	3 days ago	Indexed	1
ARE-ON Incident Docs	3 days ago	Indexed	22
CompTIA study guide	12 days ago	Error	1
Ciena Alarm Clearing 16.9	12 days ago	Error	1
CCNA study guide	12 days ago	Error	1
JNCIA	12 days ago	Error	1

Github

Total Connectors	Active Connectors	Total Docs Indexed
1	1/1	2,700

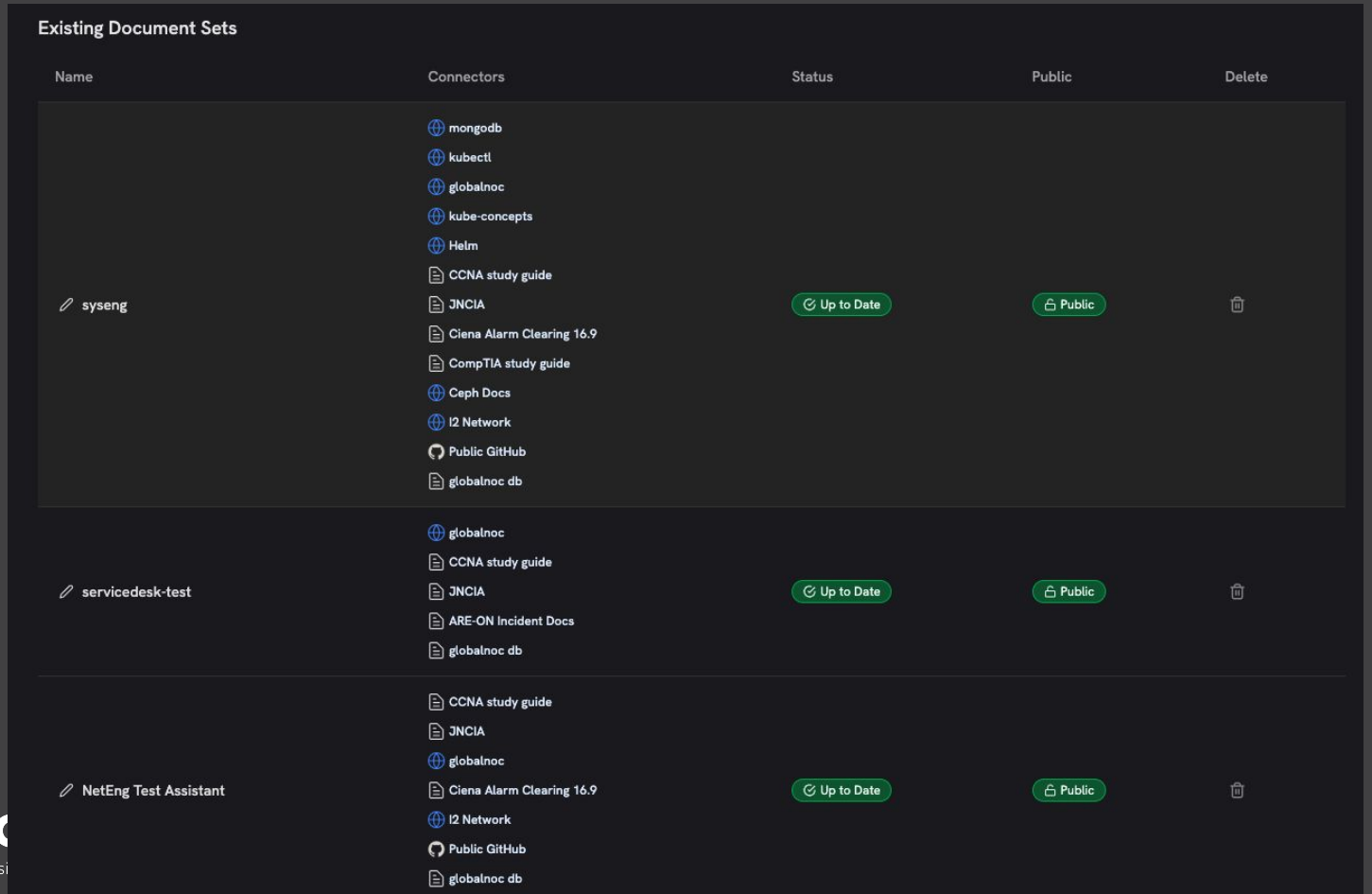
Name	Last Indexed	Status	Total Docs
Public GitHub	5 hours ago	Indexed	2700

GlobalNOC
at Indiana University

More documents

Web	Total Connectors 7	Active Connectors 7/7	Total Docs Indexed 22,456	
Name	Last Indexed	Status	Total Docs	
I2 Network	1 day ago	Indexed	37	
Ceph Docs	1 day ago	Indexed	1172	
Helm	9 hours ago	Indexed	355	
kube-concepts	8 hours ago	Indexed	175	
kubectl	8 hours ago	Indexed	162	
globalnoc	10 hours ago	Indexed	44	
mongodb	3 days ago	Indexed	20511	

Document Sets



GlobalNOC
at Indiana Universi

Assistants

Existing Assistants

Assistants will be displayed as options on the Chat / Search interfaces in the order they are displayed below. Assistants marked as hidden will not be displayed. Editable assistants are shown at the top.

Name	Description	Type	Featured Assistant	Is Visible	Delete
Syseng NOCBOT	Use this assistant to troubleshoot GlobalNOC Tools, Software, and Infrastructure	Shared	Not Featured	Visible	
ServiceDesk NOCBOT Test	Use this to test out NOCBOT functionality	Shared	Not Featured	Visible	
Internet2 Network Engineer Assistant	Internet2 Network Engineer Assistant	Shared	Not Featured	Visible	

We provided this to a few engineers

AJ Ragusa Nov 13th at 3:33 PM

@NocBot-Beta What status does a node have to be in for RANCID to run on it?

1 reply

NocBot-Beta APP Nov 13th at 3:34 PM

For RANCID to run on a node, the node must be in the "Active" status. This is because RANCID is one of the GlobalNOC tools that requires a node to be in an active state to operate on it <|[1]>.

In the context of GlobalNOC, a node's status can be one of the following: Discovered, Planning, Test, Install, Active, Occupied, Offline, or Decom. The "Active" status indicates that a node is installed in a rack and is in production [23].

The node status can affect whether a given GlobalNOC tool will operate on it. For example, RANCID will only run on nodes with an "Active" status [23].

Here is the relevant excerpt from the documentation:

"Table 45. Node Status Software Effect

Status Discov ered Planni ng Test Install Active Occup ied Offlin e Decom

RANCID No No No Yes Yes No No No

..."

[23]

Therefore, for RANCID to run on a node, the node must be in the "Active" status.

See less

👍 Helpful 👎 Not helpful

Sources:

globalnoc-database-docs.pdf

[23] Update Search Nodes | Atlas Admin API v2 d...

AJ Ragusa Aug 6th at 11:39 PM

@NocBot-Beta review incident INC0228951 and recommend remediation actions

1 reply

NocBot-Beta APP Aug 6th at 11:40 PM

I'll retrieve the details of incident INC0228951 to review the issue and recommend appropriate remediation actions. Once I have the incident information, I will analyze the problem and suggest the best course of action based on the details provided. Please hold on while I fetch the data.### Review of Incident INC0228951

Summary of Incident:

- **Incident Number:** INC0228951
- **Short Description:** Alarm - I2 node I2 Rancher: internet2: StatefulSet internet2/tsds-rabbitmq-server has not matched the expected number of replicas for longer than 15 minutes.
- **State:** In Progress (State: 3)
- **Priority:** 4 (Low)
- **Impact:** 4 (Low)
- **Urgency:** 4 (Low)
- **Severity:** 3 (Moderate)
- **Category:** Software
- **Assignment Group:** [Assigned to a specific group, details available in ServiceNow]
- **Assigned To:** [Specific user, details available in ServiceNow]
- **Status Summary:** The issue can be ignored until morning. The tsds-rabbitmq-server-0 is in a crash looping state, and logs are removed each time it restarts, making it difficult to identify the root cause at this time.
- **Next Action Date/Time:** 2025-08-06 06:25:00
- **Opened At:** 2025-08-06 06:24:59
- **Updated On:** 2025-08-06 21:01:13

Can it troubleshoot network outages?

Integrated with the GlobalNOC Network Troubleshooter

- Tries to find existing chat sessions based on the alarm and INC ticket number
  - To help with getting past context or conversations used
- Starts a new chat if one not found
- Sends the message to Onyx

```
- name: Send new message to NOCBot
  uri:
    url: "{{ ONYX_API }}/chat/send-message"
    headers:
      Authorization: "Bearer {{ ONYX_API_KEY }}"
      Content-Type: "application/json"
    method: POST
    timeout: 180
    body_format: json
    body:
      persona_id: "{{ SYSENG_NOC_BOT_ID }}"
      chat_session_id: "{{ chat_session_id }}"
      parent_message_id: null
      file_descriptors: []
      search_doc_ids: []
      retrieval_options: {}
      message:
```

GlobalNOC
at Indiana University

How it works - First Iteration Prompt

```
Please read over the JSON but do not give any wording about it.
Just use it as a reference point like a document.
Please look into ServiceNow at the INC ticket to get more context.
Please provide assumptions and reasons for what the root cause of the ticket and alarm
are.
These assumptions should be based on the ticket and raw JSON data.
Please provide steps that can be taken to remediate the alarm.
If any steps are provided, include snippets of the raw JSON command output that shows why
the step is recommended.
Please provide additional commands that the network engineer can run to get additional
context into the issue.
Do not mention the commands that were already run in the raw JSON output as additional
commands to run.
Please make the resulting answer in a format that uses headers called "Root Cause
Assumptions", "Recommended Remediation Steps", and "Recommended Commands to Run".
Please do not color or highlight any of the text.
Here is the raw JSON command output:
{{ raw_troubleshooter_output }}
```

How it works - Current Iteration Prompt

```
You are a Senior Network Engineer responsible for diagnosing and remediating network issues.
You have just received a ServiceNow incident (INC) ticket related to a network outage.

Context:
  - The ticket provides initial details about the alarm and the impacted service.
  - The following JSON contains raw command outputs collected from the affected network device.
  - Treat this JSON as your data source — it contains command names, their outputs, and metadata
about the device.

{{ raw_troubleshooter_output }}

Your Tasks:
1. Analyze the Data:
  - Review the raw JSON to identify symptoms, error messages, or anomalies.
  - Infer relevant details such as:
  - Device vendor and version (for example, from "show version")
  - Interfaces involved
  - Notable configurations, errors, or status indicators

2. Root Cause Analysis:
  - Based on the ticket and JSON, outline your assumptions about the root cause.
  - Explain why each assumption is plausible, referencing specific JSON snippets or command
sections.
```

How it works - Current Iteration Prompt

```
3. Remediation Steps:
  - Provide actionable steps to resolve or mitigate the issue.
  - For each step, include a relevant excerpt from the raw JSON that justifies the recommendation.
  - Ensure all recommended commands are appropriate for the device's vendor and OS version.

4. Recommended Commands to Run:
  - This section must include only one header titled exactly: "Recommended Commands to Run"
  - Do not include any introductory paragraphs or explanations before or after this header.
  - Start the section with a short list titled "Commands Already in JSON:" that enumerates all command names found in the raw JSON data.
  - Follow it with a short list titled "New Diagnostic Commands:" containing only commands that do not appear in the JSON data.
  - Do not add any additional subheaders (like "Further Investigation") or narrative text outside of these lists.
  - For each new command, briefly describe:
    - Why the command is being recommended.
    - What to look for in its output to help understand the issue or verify remediation.
  - Do not include any **debug** commands as that might flood the device and shut it down.
  - Have the commands be precise. If there can be a command like "show bfd sessions" but can be more specific like "show bfd sessions interface <interface>", use those.
```

GlobalNOC
at Indiana University

How it works - Current Iteration Prompt

```
Output Structure:
  The final answer must contain the following sections, and nothing else outside of them.
  Each section should use the exact header names below:

  Overview
  Root Cause Assumptions
  Recommended Remediation Steps
  Recommended Commands to Run

Style and Constraints:
  - Use concise, professional technical language.
  - Do not color, highlight, or format text beyond these headers.
  - Base all reasoning strictly on evidence from the ticket and JSON data.
  - Take into account the device vendor and version when forming conclusions and recommendations.
  - Ensure that all commands provided are correct for the specific vendor and version.
  - If you need to reference data, use short JSON snippets directly from the provided output.
```

GlobalNOC
at Indiana University

The Result!

The provided JSON data contains raw command outputs collected from the affected network device. The task is to analyze the data, perform root cause analysis, recommend remediation steps, and provide commands to run.

Data Analysis

The JSON data contains the following commands and their outputs:

- `show interfaces HundredGigE0/0/0/21`: The output shows that the interface is up, and the line protocol is up. The interface state transitions are 27, and the hardware is HundredGigE.
- `show version`: The output shows the Cisco IOS XR Software version 24.2.11 LNT.
- `show logging | include HundredGigE0/0/0/21`: The output shows logs that match on the interface, including up and down events, and BFD session state changes.
- `show controllers HundredGigE0/0/0/21`: The output shows operational data for the interface, including phy, digital optical monitoring, and statistics.

Root Cause Assumptions

Based on the provided data, the root cause of the issue appears to be related to the interface HundredGigE0/0/0/21 going up and down. The logs show that the interface changed state to Down on October 20, 20:22:04, and then changed state to Up on Oct 20, 20:22:07.

The BFD sessions on the interface also went down and up, which may be related to the interface flapping.

The probable cause of the issue is a problem with the optical signal or the interface itself.

Remediation Steps

To resolve the issue, the following steps can be taken:

1. **Check the optical signal**: Verify that the optical signal is stable and within the recommended range. Use the `show controllers HundredGigE0/0/0/21` command to check the optical signal level.
2. **Check the interface configuration**: Verify that the interface configuration is correct, including the IP address, subnet mask, and duplex settings. Use the `show interfaces HundredGigE0/0/0/21` command to check the interface configuration.
3. **Clear the interface counters**: Clear the interface counters to see if the issue persists. Use the `clear counters` command to clear the counters.
4. **Disable and re-enable the interface**: Disable and re-enable the interface to see if the issue persists. Use the `shutdown` and `no shutdown` commands to disable and re-enable the interface.

Lots more work to do!

If we integrate more knowledge about the network into Onyx (like understanding the network topology and hierarchy) we can probably get it to take all the alarm data and correlate it to a specific portion of the network (ie... layer 1 network) and have it troubleshoot that automatically. We have a lot of work to get there though.

GlobalNOC
at Indiana University

Thanks!

aragusa@globalnoc.iu.edu