

# NSX ConfigGen

## 1. Introduction:

NSX ConfigGen is a user-friendly tool designed to automate NSX configuration tasks by generating Python or Terraform scripts based on user input and knowledge of NSX APIs. This document outlines the design principles, architectural components, and functional flow of NSX ConfigGen.

## 2. Overview:

NSX ConfigGen offers a conversational interface, similar to a chatbot, to guide users through the process of specifying their desired NSX configurations. By leveraging NLP and its understanding of NSX APIs, the tool translates user requests into executable scripts, eliminating the need for manual scripting and minimizing configuration errors.

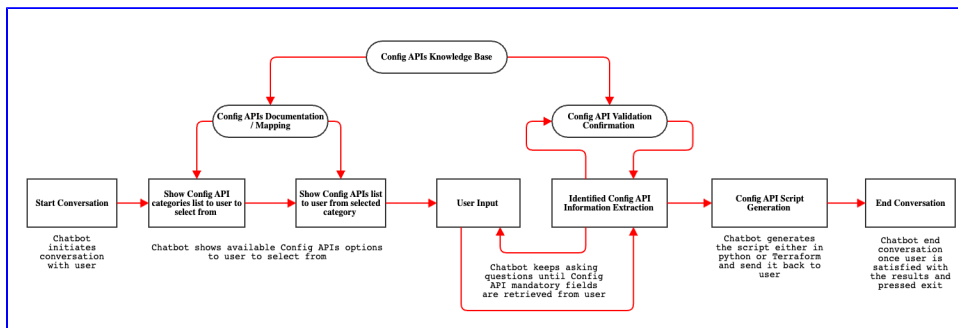
## 3. Architecture:

### 3.1. Components:

- **Conversational Interface:** The chat-like interface acts as the primary user interaction point, capturing user requirements and preferences.
- **Intent Understanding:** NLP techniques analyze user input, and identify the intended configuration operation (create, update, delete) and the target NSX component.
- **Config APIs Knowledge Base:** An internal repository that stores comprehensive information about NSX APIs, including API schemas, resource types, properties and its types, etc.
- **Script Generator:** Based on the user's intent and extracted configuration details, the script generator creates Python or Terraform scripts to execute the desired operation.
- **Error Handling:** A robust system handles invalid inputs, API errors, and potential conflicts, providing informative feedback to the user.

### 3.2. Data Flow:

1. **User Input:** The user interacts with the chatbot, describing their desired configuration.
2. **Intent Recognition:** NLP analyzes the user's input, extracting the intended operation and target NSX component.
3. **API Mapping:** The API knowledge base identifies the relevant API call and required parameters for the requested configuration.
4. **Parameter Extraction:** The system prompts the user for specific configuration details corresponding to the identified API call parameters.
5. **Script Generation:** The script generator utilizes the extracted information and API knowledge to create a Python or Terraform script for the intended operation.
6. **Script Presentation:** The generated script is presented to the user for review and confirmation.
7. **Execution and Feedback:** Upon user confirmation, the script can be executed against the NSX environment, with feedback provided on its status and results.



## 4. User Interaction:

The chatbot interface offers a flexible and intuitive means for users to interact with NSX ConfigGen. It caters to both new and experienced users, offering a balance between guided prompts and open-ended commands. Features include:

- **Menu-driven navigation:** Users can explore available configuration options through a structured menu system.
- **Natural language input:** Users can express their desires in natural language, allowing for ease of use.
- **Contextual prompts:** Dynamic prompts adapt to the chosen configuration path, guiding users through essential parameter collection.
- **Help and feedback:** An integrated help system assists users with unfamiliar options and provides avenues for feedback.

## 5. Benefits:

- **Increased Efficiency:** Automates repetitive configuration tasks, saving time and effort.

- **Reduced Errors:** Minimizes manual configuration errors, improving accuracy and consistency.
- **Accessibility:** Provides a user-friendly interface for both novice and expert NSX users.

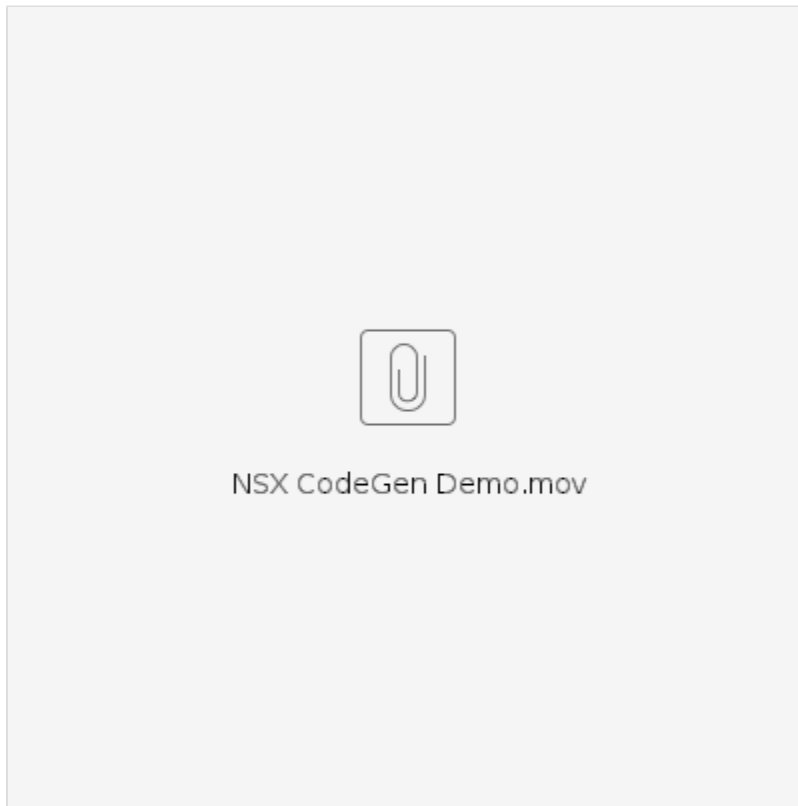
## 6. Further Enhancements:

- **Testing and validation framework:** Streamlines script testing and verification.
- **Advanced NLP capabilities:** Refines intent recognition and user interaction for personalized experience.
- **API documentation integration:** Connects directly to NSX API documentation for reference.
- **Multi-language support:** Expands accessibility to users with varied script preferences.

## 7. Conclusion:

NSX ConfigGen stands as a valuable tool for streamlining NSX configuration management, automating tasks, and reducing human error. With its combination of NLP, comprehensive API knowledge, and user-friendly interface, NSX ConfigGen empowers users to efficiently and accurately manage their NSX environments.

## 8. Demo:



<https://drive.google.com/file/d/1uQvoZoX3g6XVV9FkakZOI-I0C!gcPW8I/view?usp=sharing>

Also, this bot could not answer any questions that are not related to NSX

## NSX CodeGen

Assistant: "Hello! I'm here to help you create a distributed firewall rule. Could you please provide me with the details of the rule you want to create?"

Could you please help me book tickets on make my trip

I'm sorry for the confusion, but as an NSX Config Generator AI assistant, I'm specialized in generating NSX configuration scripts based on user requirements. I'm afraid I can't assist with booking tickets on Make My Trip. However, I'd be more than happy to help with any questions or tasks related to NSX configurations. Could you please share your requirements in that regard?

END CONVERSATION