# LLM API Security Demo – Lab Setup Guide

This guide provides the prerequisites and step-by-step setup instructions for running the OLLAMA API security demo on a cloud-hosted Windows VM. Follow these steps to recreate the lab environment used in this course.

This lab demonstrates the risk of exposing LLM APIs to the public internet without authentication, as discussed in Section 3 of the course. By the end of this lab, you'll understand how remote users can interact with, manipulate, and even delete models if proper security controls are not in place.

## Demo Prerequisites

- A cloud VM (Azure or AWS) – Windows OS preferred
- OLLAMA API & Framework – Download from https://ollama.com
- AI Model – Choose a model like LLaMA2 or Mistral from the Ollama store
- Open firewall rule for port 11434 – Required to expose OLLAMA API externally
- A laptop with RDP access – Used for managing the lab VM and testing externally

## Step-by-Step Setup (Windows + Azure)

1. Go to Azure Portal and create a new Windows VM (use Free Trial or Pay-As-You-Go plan).

2. Choose a region and a size suitable for testing (B1s is usually enough).

3. After deployment, connect via RDP to the public IP.

4. Open browser and download OLLAMA from https://ollama.com.

5. Install OLLAMA and run the model using:

**You can run: ollama run tinyllama or replace tinyllama with any model available from Ollama's library.**

> ***`ollama run llama2`***

6. Ensure OLLAMA is listening on all interfaces (0.0.0.0) in its config. Bind Ollama to listen on all interfaces: ollama serve --host 0.0.0.0

**Use netstat -ano | findstr 11434 to verify that Ollama is listening on all interfaces.**

7. Go back to Azure > Networking > Add inbound port rule for TCP 11434.

8. From your local machine, run:

> ***`curl http://<public-ip>:11434/api/tags`***

This should return the list of models if the port is exposed properly.

## Remote Access Testing Using Curl

Interact with OLLAMA API Remotely (Advanced Commands)

1. Query API to list available models:

```
curl http://<public-ip>:11434/api/tags
```

2. Interact with the model (send a prompt):

```
curl -X POST http://<public-ip>:11434/api/generate \
-H "Content-Type: application/json" \
-d '{
  "model": "tinyllama",
  "prompt": "Explain LLM security",
  "stream": false
}'
```

3. Delete a model from the remote OLLAMA server:

```
curl -X DELETE http://<public-ip>:11434/api/delete \
-H "Content-Type: application/json" \
-d '{
  "name": "tinyllama"
}'
```

4. Pull (download) a model remotely to the OLLAMA server:

```
curl -X POST http://<public-ip>:11434/api/pull \
-H "Content-Type: application/json" \
-d '{
  "name": "tinyllama"
}'
```

## Security Tips

- Never expose OLLAMA API to the internet without authentication.

- Use NGINX or any reverse proxy to wrap OLLAMA with basic auth or JWT.

- Monitor your firewall rules and disable unused ports.

## Clean-Up Tips (After Lab)

- Revert firewall rule if you're done testing.

- Stop Ollama or rebind it to localhost (127.0.0.1) to prevent unintended exposure.

- Remove any exposed models if you're using a shared cloud environment.

## Note:

Once you’ve completed the steps in this guide, try sending prompts to your model, run the delete and pull commands, and observe the system’s behavior. Then try the same setup with NGINX authentication in place, as shown in the next demo. This will give you a full before-and-after understanding of the risk.