**Deployment Manual for OAUTH Server**

**Scope:**

OAUTH server defines the mechanism to limit an application’s access to a user’s account data. An application can request data with an issued access token to the application. The application request will be presented to the server, server will be granted data according to the scope defined. The issued access token will expire after a limited time. This increases the atomic security of user data.

**Requirements:**

The following are the hardware requirements of the OATH Server:

* CPU that can manage 2 threads/process at a time
* 4 GB RAM
* 8 GB Hard Disk
* 2 MB/s constant Internet Connection

The following are the software requirements of the OATH Server:

* Linux Operating System
* Python 3
* Django
* Django Rest Frame Work
* Oauth provider

**Step-by-Step Instructions:**

Following are the instructions for the complete deployment:

* Clone the GitHub repository of the project using this link: [OAUTH Server](https://github.com/EpiVision/Django-Module-Project.git)
* The project contains a file (this file contains the command required to install libraries of the project)

*Django-Module-Project > moduleProject > liberaries\_for\_server\_deployment - Copy.txt*

Navigate to the file and run this command

pip install -r ‘file\_Name\_provided\_above’

* Make sure all libraries are installed. If any library-issued error, go and check the latest version of the library at your time of deployment.
* Now you are ready to run your deployment.

**Run Server:**

Now it’s time to run the server on the Internet. Navigate to file

*Django-Module-Project > moduleProject > manage.py*

Type this command on the console:

python3 manage.py runserver “0.0.0.0:8000”

0.0.0.0 IP address and 8000 port are used as parameters to run the server. This parameter allows the system to accept requests on all available public IPs using port 8000. After this command's successful execution, the server would start and would be ready to accept requests from the internet. The running server can be visualized below:

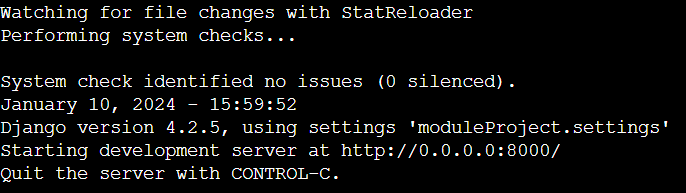


Figure 1: Console on Running Server

**Testing & Validation:**

The server can be tested as it is running as per requirement or not. The simplest way to test a hosted system is to call API. OAUTH server API can be called using a test user. Create a test user named ‘abc’ using the Django admin panel with password abc123. The server running IP address is let's say x.x.x.x:8000. Use a link <http://x.x.x.x:8000/o/token/> (replace x.x.x.x with public IP of your machine) with username abc, password abc123, and grant\_type password in the body of the request. Under the Authorization tab, select Basic Auth in the Type drop-down and fill hashed username and password that the Django admin panel provides you. Send the request using the POST type. This whole test is completed using the Postman tool.

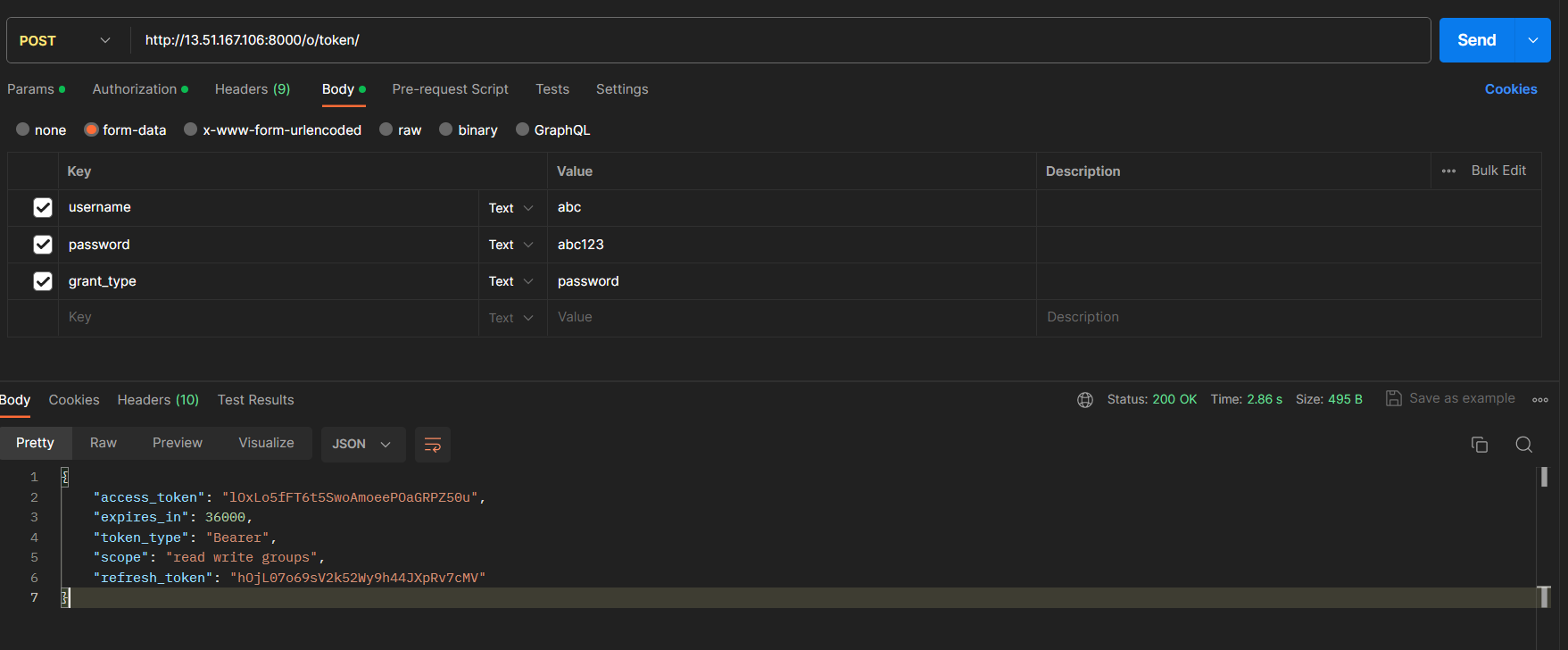


Figure 2: OAuth Server testing with test user using Postman

This is the successful call of API with some attribute in response like access\_token, expires\_in, etc. (every attribute has some meaning like access token means all data related to user abc would be accessed using this access token and this access token is valid for the time mentioned in seconds in attribute expires\_in etc.).

Now make another API call for accessing the data of the user abc using access\_token in the above response. Use the link <http://x.x.x.x:8000/api/protected-resource/> (replace x.x.x.x with your machine's Public IP address) with GET type and paste access\_token from the above response in Barer Token under the Authorization tab. This can be visualized using the image given below:

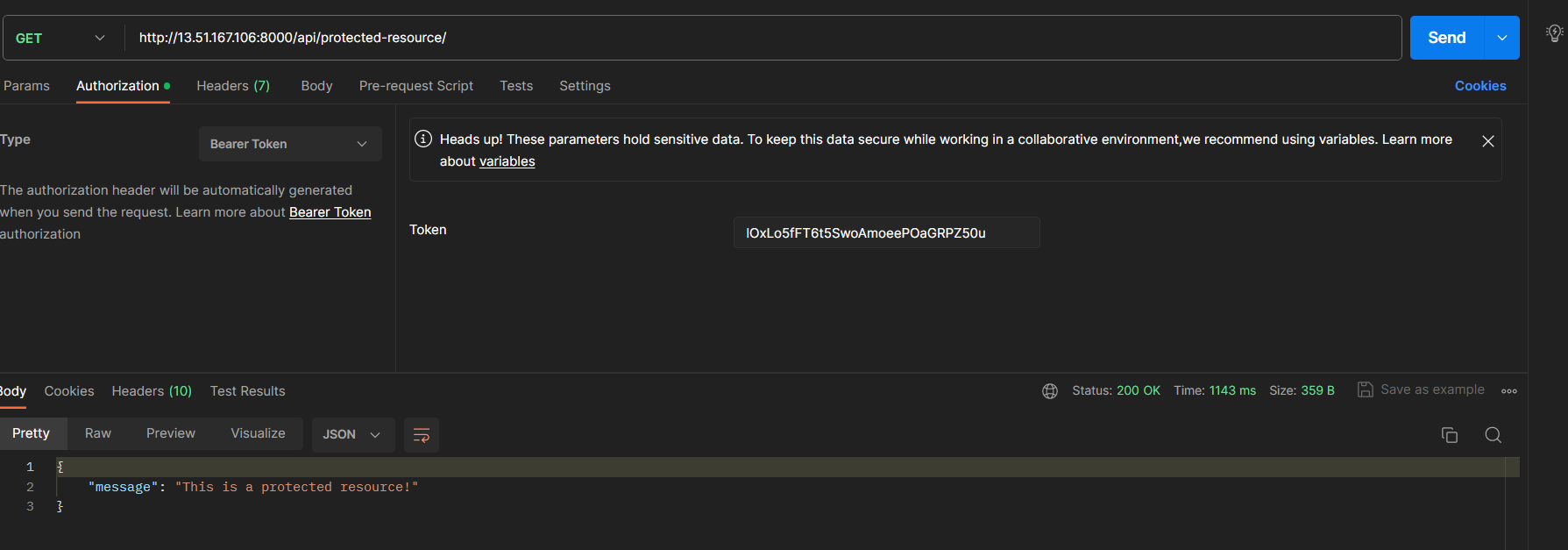


Figure 3: Access to protected data of user abc

**Troubleshooting & FAQs:**

Following are some common problems with their solution that users may face during the deployment phase.

**Q1: I've deployed an OATH server on the machine, but I'm unable to access it. What could be the issue?**

Possible issues and troubleshooting steps:

* **Security Groups:** Check the machine security group settings to ensure the necessary ports (e.g., 80 for HTTP or 443 for HTTPS) are open for incoming traffic.
* **Firewall Configuration:** Verify that the server's firewall (such as iptables) allows incoming connections on the required ports.
* **Network ACLs:** Ensure that the Network Access Control Lists (ACLs) associated with the subnet allow traffic to and from the instance.
* **Public IP/Hostname**: Confirm that you're using the correct public IP or hostname to access the server.

**Q2: After deploying the OATH server, it's showing a 404 error when accessing the web interface. What steps can I take to troubleshoot this?**

Possible issues and troubleshooting steps:

* **Web Server Configuration:** Check the web server configuration (e.g., Apache, Nginx) to ensure that the server's document root and application paths are correctly set.
* **Application Deployment:** Verify that the OATH server application has been deployed and started successfully.
* **File Permissions:** Ensure that the necessary files and directories have appropriate permissions for the webserver to access them.
* **Restart Services:** Restart the web server and associated services to ensure any configuration changes take effect.

**Q3: I've deployed the OATH server, but it's not communicating properly with the authentication database. How can I troubleshoot this connectivity issue?**

Possible issues and troubleshooting steps:

* **Database Configuration:** Check the configuration settings of the OATH server to ensure correct database credentials and connection settings.
* **Database Access:** Verify that the machine has network connectivity to the database server and that no firewall or security group restrictions are blocking the connection.
* **Database Service Status:** Ensure that the database service is running and accessible from the machine.
* **Database Logs:** Check the logs of both the OATH server and the database server for any error messages related to failed connections or queries.

These questions cover various common issues one may encounter when deploying an OATH server on a machine and provide steps to troubleshoot and resolve them.

**Support & Contacts:**

This project is completed on an EC2 instance of AWS. You might face other issues that are not covered above due to a change of environment. In case of any issue, you can email <muhammadalimurtaza997@gmail.com>

*Best of Luck*