Exposé vos services en entreprise à la Cloud API

Tous les appels API qui sortent vers vos services en entreprise passent par l'API Gateway de la Cloud. L'API Gateway gère les demandes des utilisateurs, l'authentification des utilisateurs via OAuth, l'enforcement des politiques de sécurité, etc.

Lorsque vos services en entreprise sont privés de votre Intranet, l'API Gateway ne peut pas les accéder via Internet. Par conséquent, vous devez exposer vos services internes aux utilisateurs publics de sorte qu'ils puissent être accessibles par l'API Gateway.

**Dans cette session**, vous apprendrez comment exposer vos services aux utilisateurs publics.

**Let's get started!**

- Exposez vos services via un serveur DMZ
- Exposez vos services via un tunnel VPN

## Exposez vos services via un serveur DMZ

Vous pouvez placer un proxy en zone démilitarisée (DMZ), et connecter vos services en entreprise à celui-ci. Tous les appels de vos services en entreprise seront dirigés vers le proxy. Les étapes ci-dessous expliquent comment faire cela en utilisant un exemple de service et NGINX comme proxy.

**Before you begin,**

- Exposez vos services via un serveur DMZ
- Exposez vos services via un tunnel VPN

**Securisez vos services** comme ils seront accessibles publics apres avoir exposé vos services via le proxy.

**Set up NGINX** dans la DMZ de votre Intranet. Voir les instructions d'installation de NGINX et les commandes de base.

**Tip:** bien que les captures d'écran donnés ici peuvent varier en fonction de votre API et de la configuration de votre service, vous pouvez suivre les mêmes instructions. Nous utilisons NGINX comme proxy, mais vous pouvez également utiliser toute autre technologie de manière similaire.

1. **Go to the <NGINX_HOME>/sites-enabled folder and create a VHost file.** Il rute les appels qui entrent à NGINX avec le nom d'hôte de votre service interne.
2. **Add the following in the config file that you created in step 1:**
   - Create an upstream and point it to the IP et port of the actual backend service.
   - Inside the `server()` block, give the name of the server and the port that the requests are coming from. In this case, it is `mycompany.services.com`.
   - Inside the `location()` block, route a request to the actual backend service if the request comes in the pattern of the server name mentioned above.
upstream myservice {
    server <IP_of the backend_service>:<port_of the backend_service>;
    #For example, server 10.5.10.70:9443;
}
server {
    listen 80;
    server_name mycompany.services.com;
    location / {
        include /etc/nginx/proxy_params;
        proxy_pass https://myservice/;
    }
    access_log /mnt/var/log/nginx/mycompnay/access.log;
    error_log /mnt/var/log/nginx/mycompany/error.log debug;
}

3. Save and reload your NGINX configuration using the following command:

```
service nginx reload
```

Next, let's make sure that the host that we mentioned in the above VHost config (i.e., mycompany.services.com) publicly resolves to the IP address of the reverse proxy server. If not, when you call this endpoint from WSO2 API Cloud, it will not be able to identify the location correctly.

4. In an available DNS server, map the IP of your NGINX with the domain name that you specified as the server_name (i.e., mycompany.services.com).

```
10.5.10.49 -> mycompany.services.com
// 10.5.10.49 is the IP addresses of the server where placed your reverse proxy
```

Once you have done the required configs in the NGINX reverse proxy, test whether a call is correctly routed through the NGINX to your backend services.

5. Send the following Curl request using the command-line or the Terminal.

In this example, the backend is secured using basic auth. Therefore, we pass the `<base64-encoded username:password>` in the Authorization header.

```
curl -k -X GET --header "Authorization: Basic <base64-encoded username:password>" http://mycompany.services.com/jaxrs_basic/services/customers /customerservice/customers/123
```

6. Check whether you get a result from your actual backend service. If so, your reverse proxy configuration has been done correctly.

Now that the configurations are complete, let's design the API using the API Cloud.
7. Log in to WSO2 API Cloud and start to create a new API by clicking the **ADD NEW API** link. Alternatively, you can edit an existing API.

8. In the **Design** tab of the API, under the **API Definition**, give the URI template that matches the resources of your backend service. In this example, the backend requires you to pass a customerId. So, the URL pattern is `jaxrs_basic/services/customerservice/customers/{customerId}`.

9. In the **Implement** tab, specify the endpoint that will be resolved at your reverse proxy before calling your backend service. According to the reverse proxy configuration in this example, the endpoint is `http://mycompany.services.com/`.

10. Since the backend service is secured using basic authentication in this example, set the **Endpoint Security Scheme** to **Secured**, the **Endpoint Auth Type** to **Basic Auth** and give the credentials expected by the backend service.

11. In the **Manage** tab, select all the available tiers and save and publish your API.
12. Go to the API Store, subscribe to the API and click its API Console tab using which you can invoke the API.

13. Give a customerId (say 123), invoke the API and note the response that is displayed. The API Gateway makes a call to your reverse proxy, which is resolved using the DNS mapping. The reverse proxy then calls your backend service and returns the response to the API Console.

With the help of a DMZ server, you have invoked a backend service that is private to your Intranet using an API in WSO2 API Cloud.

**Expose your services using a VPN**

If you are unable to use a reverse proxy, you have the option to create a VPN connection between your network and WSO2 API Cloud.

*Tip:* Make sure you Secure your backend services as there is capability for someone else also to invoke your backends through the cloud API gateway after you expose them via the VPN.

This solution is fully secured and managed.

- Each customer gets a separate subnetwork in the WSO2 Cloud space.
- The subnetwork has a highly available load-balancer cluster, which connects to your network via AWS VPN.
- To get started, click the Support menu in API Cloud interface and submit your request. WSO2 will respond and guide you through the setup process.