

BE360 – WhatsApp for Business - VONAGE

This guide contains a step-by-step procedure that BeCloud's Agents must follow to obtain the integration between the Agent Desktop and WhatsApp for Business.

Warnings before reading:

- The "Provider" mentioned many times in this guide was Vonage as of June 2023, so each example uses the Vonage environment and methods.

Some requirements must be met by the customer before starting the integration:

1. Meta verified Business Account and Business Manager
2. A number that can be reachable from the USA
3. *Access to the customer Meta Business account and being a Facebook Administrator
4. Verified Facebook Account

If the requirements are met, a BeCloud Agent must send the customer a checklist that contains the sum of information that a BeCloud Agent will overwrite on the provider's [WABA form activation](#) and (if needed) also the [WABA form activation for multiple numbers](#).

After some days, the customer will receive a request for the "Jewel" on their Meta Business account on the notification panel, which must be accepted.

Now it's time to create the new environment for the customer's WABA, but first, it is necessary to obtain a new API Key, to do so:

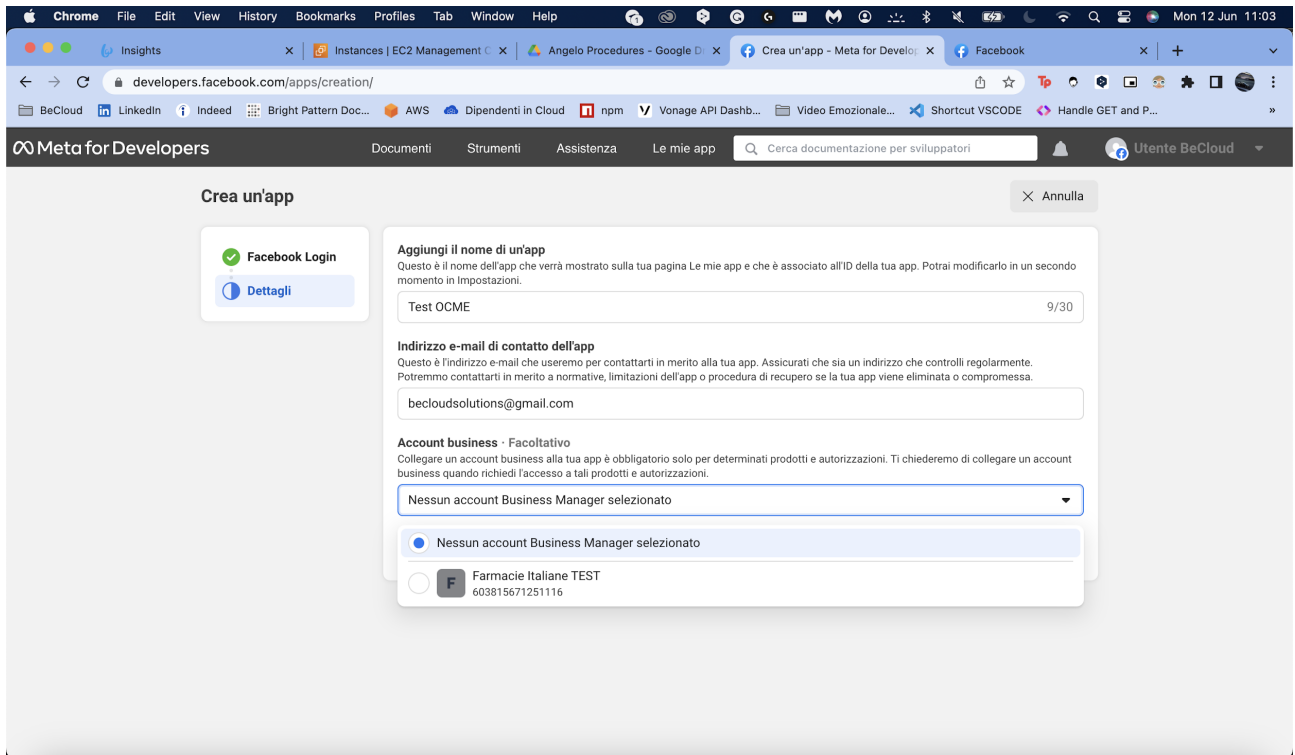
1. Access the provider's Dashboard
2. Go to API Settings > Request new API Keys
3. Fill out the form requesting a new API Key for your new environment

After obtaining the new API Key, the new environment must be set up with the new script that all the API messages will be routed.

*Keep in mind that, to create **more than two WhatsApp channels**, it is necessary to be an admin of the Facebook Meta page of the company that required the service. Because Meta (Facebook) **requires a verified Business** to create more than **two channels** and **two numbers**.

A BeCloud agent must create an App on the Facebook Developer's page. In order to make changes in the process.

To do so, follow the [following link](#) and follow the wizard:



At this point you must select the Account Business of the customer that gave you the admin privileges.

SSH – Be360WhatsApp Instance

On the Be360WhatsApp Instance create a new folder named be360customer. Inside the same folder, create the new package.json (there is information about who created the new 'server' and all the installed packages), for doing so, type on the bash 'npm init' and follow the wizard.

Copy "app.js", "main.js" and "constants.json" from one of the folders in production and move them to the new folder, create as well the subfolders (files, filesIn, and past_logs) and the log file (empty).

Install required libraries through "npm" (list of used libraries resides at the top of each ".js" file after the "require"): *"npm install **packet_name**"*

If the terminal cannot access the file you've created, set the folder as writable to anyone with this command: *"chmod -r 777 **be360test**"*.

Edit "constants.json" with correct pieces of information regarding authentication (Provider and BP sides):

- "bpdomain": "**tenant_url**",
- "bptenant": "**tenant_name**",
- "bpappid": "**app_id retrieved from chat scenario**",
- "nexmotype": "**WhatsApp**" (specifies in what platform the message will be sent),
- "nexmonumber": "**phone number of whatsapp integration**",
- "nexmotoken": "retrieved by a conversion in base 64 of these parameters (**api_key:api_secret**)",
- "nexmoapikey": "**api_key**",
- "nexmosecret": "**api_secret**",

- "nexmo_url": "<https://api.nexmo.com/v0.1/messages/>",
- "localfilepath": "<http://wabX.ccaas.becloudsolutions.com/files>"

Example:

- "bpdomain": "beincontact.becloudsolutions.com",
- "bptenant": "TENANT_EXAMPLE.it",
- "bpappid": "haff3284239482shidf28y934",
- "nexmotype": "whatsapp",
- "nexmonumber": "393518652380",
- "nexmotoken": "HADgaosfayAY)(£")71712789==",
- "nexmoapikey": "ja92fu838n",
- "nexmosecret": "dsfguuf912737128941jsma",
- "nexmourl": "<https://api.nexmo.com/v0.1/messages/>",
- "localfilePath": "<http://wabfarmacie.ccaas.becloudsolutions.com/files/>"

At the bottom of each "app.js" it is necessary to change the port (check the Nginx config file to verify which ports are already in use, it is suggested to use ports above 5000 in up) and open each port in the security group.

Create a new domain in Route53 and add it to the Nginx configuration as well.
(Nginx configuration resides in </etc/nginx/conf.d/node.conf>:

```
• server 127.0.0.1:500X max_fails=0;
• server {
•     listen 80;
•     server_name test.ccaas.becloudsolutions.com;
•     location / {
•         proxy_set_header X-Real-IP $remote_addr;
•         proxy_set_header X-Forwarder-For $proxy_add_x_forwarded_for;
•         proxy_set_header Host $http_host;
•         proxy_redirect off;
•         proxy_pass http://127.0.0.1:500X;
•     }
• }
```

At the end of this configuration, type in the bash "restart nginx" to validate the changes.

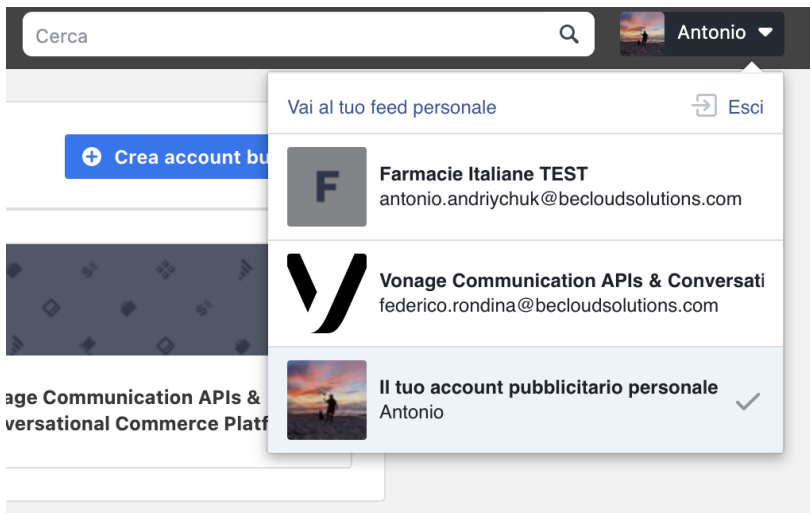
It is necessary to set up the listener for the new application, for doing so:

start the new application using the pm2 node process inside the folder you've created with this command:
*pm2 start app.js --name **test** --log wab.log*. Test that everything is working and process stays up by having multiple concurrent chats.

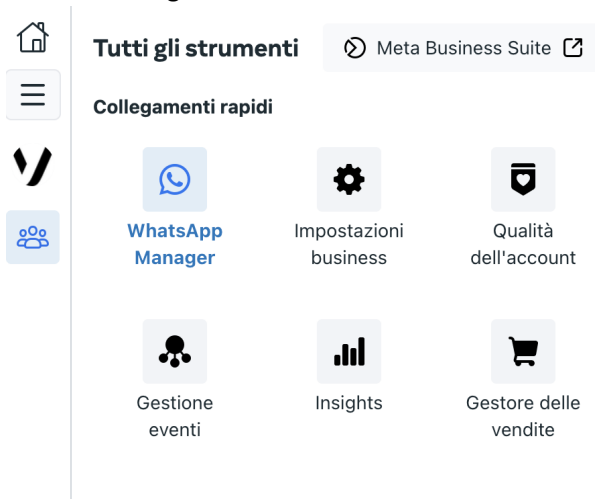
Log in to WABA Dashboard & Create Message Templates

After the above steps are completed, your users will now be able to log in to your **WhatsApp Business Account Dashboard** by following the steps below:

1. Log into your Facebook business and visit your [business settings page](#)
2. Make sure that you are on the Nexmo business page
 1. In the top right corner, there is a drop-down option, select Nexmo
 2. See the screenshot here for help



3. In the left-hand column, click on "WhatsApp Accounts" under the Accounts section
4. Then click on your Company WABA, then under the Settings tab, you can click on "WhatsApp Manager"



5. The Insights, Messages Templates, and Phone Numbers tabs should now be seen.

6. Under the Templates tab, you can create the message templates you want to use

Each template will need to be approved by Facebook, which normally takes between 24-48 hours, so we recommend creating these sooner rather than later.

Create Messages and Dispatch APP

See the steps below for reference on how to create a Messages and Dispatch app:

1. Navigate to your Provider's dashboard.
2. Click on "Application" from the left sidebar
3. Create a new application and follow the wizard:
 - Create a name for the application.
 - Generate a new public and private key, and save the key in a safe folder on your device.
 - On the Messages provide the Inbound url and status url, formatted as:
 - o Inbound URL: <http://wabtenant.ccaas.becloudsolutions.com/webhooks/inbound-message>
 - o Status URL: <http://wabtenant.ccaas.becloudsolutions.com/webhooks/message-status>

- Generate the new application
- 4. Select “External Accounts” from the left tab
- 5. Below the WhatsApp icon click on “[Set up my WhatsApp Business Account](#)”
- 6. Follow the wizard, connect on the Meta account that gave you the admin privilege
- 7. At a certain point of the Wizard you will be asked to provide a phone number:
 - Write the phone number that will be linked to the WhatsApp service
 - After filling the form with the information from the Checklist (About, Name of the WhatsApp Account, description, link, etc.), the next step is to verify the number.
 - To verify the number, you must be connected to the agent desktop and ensure all the calls can be routed to your softphone/user . The number will be nor American (+1) nor British (+44) so be sure that these kinds of calls will be routed to your user.
 - Accept the incoming call and be sure to write down the code to verify the account.
- 8. After verifying the account, simply edit the WhatsApp account as requested on the Check List and link your application to the external account.

What’s left to do is just create the chat scenario and test.