



User Instructions: Setting up Visual Studio

Launch the product via 1-click. **Please wait until** the instance passes all status checks and is running. You can connect using your Amazon private key and 'ubuntu' login via your SSH client.

To update software, use: **sudo apt-get update**

1. Configure NGINX to your instance IP address. Run:

sudo nano /etc/nginx/sites-available/my_project

Replace the **server_name** with your **Instance Public IPv4 address**

```
GNU nano 6.2 /etc/nginx/sites-available/my_project
server {
    listen 80;
    server_name 34.201.54.201;

    root /usr/local/bin/my_project/public;
    index index.php;

    location / {
        try_files $uri /index.php$is_args$args;
    }

    location ~ ^/index\.php(/|$) {
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock; # Adjust if using a different PHP version
        fastcgi_split_path_info ^(.+\.php)(/.*)$;
        include fastcgi_params;

        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param DOCUMENT_ROOT $document_root;

        internal;
    }

    location ~ \.php$ {
        return 404;
    }

    error_log /var/log/nginx/your_project_error.log;
    access_log /var/log/nginx/your_project_access.log;
}
```

- **Save & Exit**

2. Reload Ngnix, run:

```
sudo systemctl reload nginx
```

3. Check NGINX status

```
sudo systemctl status nginx
```

4. Change into the Symfony directory:

```
cd /usr/local/bin
```

5. Here you will find all the installed applications.

```
ls -la
```

6. Confirm all requirements, run:

```
symfony check:requirements
```

```
Symfony Requirements Checker
=====

> PHP is using the following php.ini file:
/etc/php/7.4/cli/php.ini

> Checking Symfony requirements:

.....

[OK]
Your system is ready to run Symfony projects

Note The command console can use a different php.ini file
~~~~ than the one used by your web server.
Please check that both the console and the web server
are using the same PHP version and configuration.
```

7. Log into the project folder:

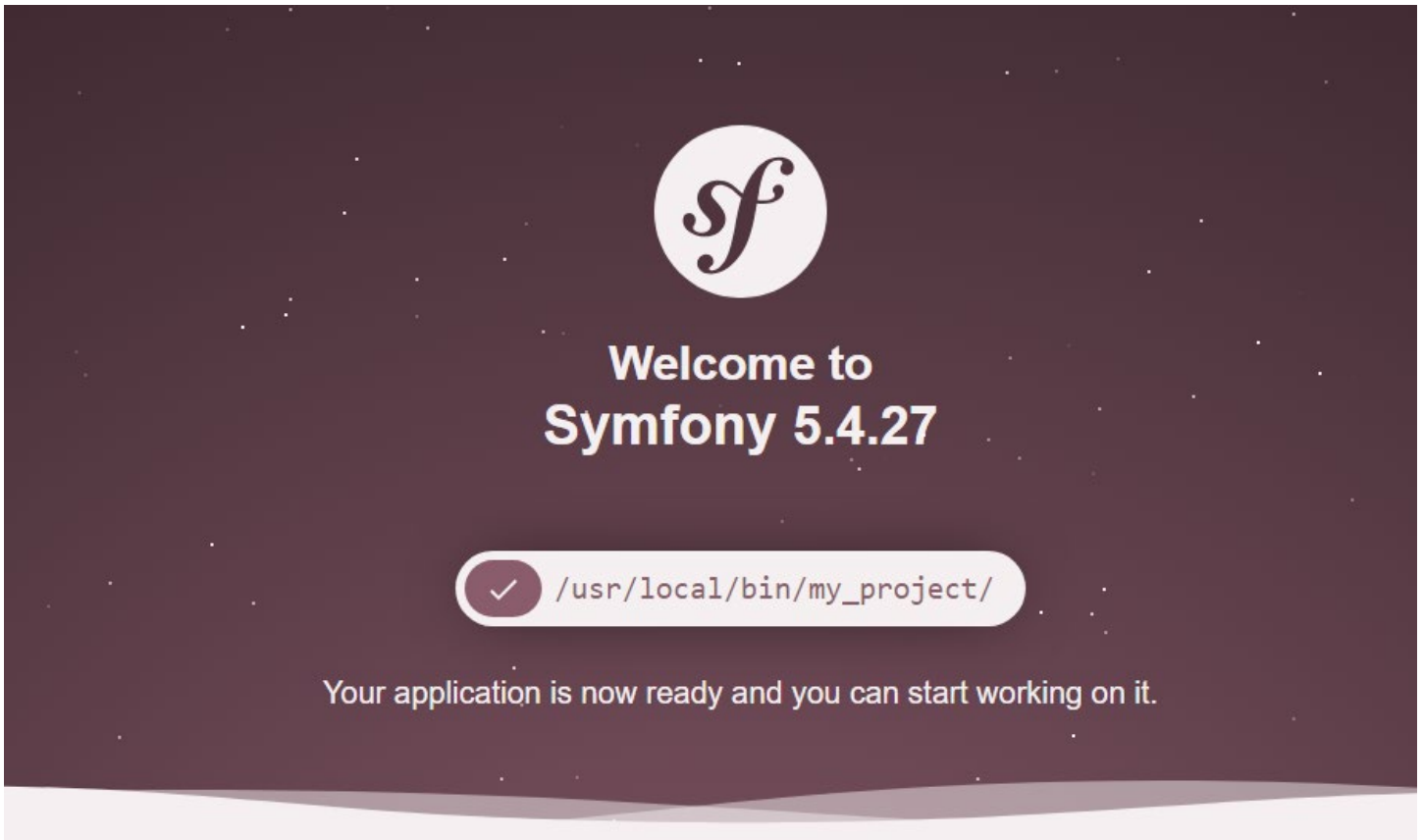
```
cd my_project
```

```
(cd /usr/local/bin/my_project)
```

8. Start Symfony by running:

symfony

9. In browser, go to: **http://Your_Instance_Public_IP** to test settings. You should see the Symfony web page:



- **Get started in Symfony:** <https://symfony.com/doc/current/index.html>

AWS Data

- Data Encryption Configuration: This solution does not encrypt data within the running instance.
- User Credentials are stored: /root/.ssh/authorized_keys & /home/ubuntu/.ssh/authorized_keys
- Monitor the health:
 - Navigate to your Amazon EC2 console and verify that you're in the correct region.
 - Choose Instance and select your launched instance.
 - Select the server to display your metadata page and choose the Status checks tab at the bottom of the page to review if your status checks passed or failed.

Extra Information: (Optional)

Allocate Elastic IP

To ensure that your instance **keeps its IP during restarts** that might happen, configure an Elastic IP. From the EC2 console:

1. Select ELASTIC IPs.
2. Click on the ALLOCATE ELASTIC IP ADDRESS.
3. Select the default (Amazon pool of IPv4 addresses) and click on ALLOCATE.
4. From the ACTIONS pull down, select ASSOCIATE ELASTIC IP ADDRESS.
5. In the box that comes up, note down the Elastic IP Address, which will be needed when you configure your DNS.
6. In the search box under INSTANCE, click and find your INSTANCE ID and then click ASSOCIATE.
7. Your instance now has an elastic IP associated with it.
8. For additional help: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html>