NGINX Buildpack

This topic describes how to push your NGINX app to Cloud Foundry and how to configure your NGINX app to use the NGINX buildpack.

Push an App

If your app contains an `nginx.conf` file, Cloud Foundry automatically uses the NGINX buildpack when you run `cf push` to deploy your app.

If your Cloud Foundry deployment does not have the NGINX buildpack installed or the installed version is outdated, run `cf push YOUR-APP -b https://github.com/cloudfoundry/nginx-buildpack.git` to deploy your app with the current buildpack. Replace `YOUR-APP` with the name of your app.

For example:

```
$ cf push my-app -b https://github.com/cloudfoundry/nginx-buildpack.git
```

Configure NGINX

We recommend that you use the default NGINX directory structure for your NGINX webserver. You can view this directory structure in the nginx-buildpack repository in GitHub.

The NGINX webserver setup includes the following:

- A root folder for all static web content
- A MIME type configuration file
- An NGINX configuration file
- A `buildpack.yml` YAML file that defines the version of NGINX to use. As an example, see buildpack.yml in the Cloud Foundry NGINX Buildpack repository in GitHub.

You should make any custom configuration changes based on these default files to ensure compatibility with the buildpack.

Create the nginx.conf File

Use the templating syntax when you create an `nginx.conf` file. This templating syntax loads modules and binds to ports based on values known at launch time.

Port

Use `{{port}}` to set the port to listen on. At launch time, `{{port}}` will interpolate in the value of `$PORT`.

Note: You must use `{{port}}` in your `nginx.conf` file.

For example, to set an NGINX server to listen on `$PORT`, include the following in your `nginx.conf` file:

```
server {
  listen {{port}};
}
```

Environment Variables

To use an environment variable, include `{{env "YOUR-VARIABLE"}}`. Replace `YOUR-VARIABLE` with the name of an environment variable. At staging and at launch, the current value of the environment variable is retrieved.

For example, include the following in your `nginx.conf` file to enable or disable GZipping based on the value of `GZIP_DOWNLOADS`:

```
gzip {{env "GZIP_DOWNLOADS"}};
```

- If you set `GZIP_DOWNLOADS` to `off`, NGINX does not GZip files.
- If you set `GZIP_DOWNLOADS` to `on`, NGINX GZips files.

Unescaped Environment Variables

To use unescaped environment variables, add an array of environment variable names to the `buildpack.yml`. See the following example:

```
---
nginx:
  version: stable
  plaintext_env_vars:
    - "OVERRIDE"
```

In this example, the `OVERRIDE` environment variable can contain `.json` content without being `html` escaped. Ensure that you properly quote such variables to appear as strings in the `nginx.conf` file.

Loading Dynamic Modules

To load an NGINX module, use the following syntax in your app's `nginx.conf` file:

```
{{module "MODULE-NAME"}}
```

If you have provided a module in a `modules` directory located at the root of your app, the buildpack instructs NGINX to load that module. If you have not provided a module, the buildpack instructs NGINX to search for a matching built-in module.

As of v0.0.5 of the buildpack, the `ngx_stream_module` is available as a built-in module.

For example, to load a custom module named `ngx_hello_module`, provide a `modules/ngx_hello_module.so` file in your app directory and add the following to the top of your `nginx.conf` file:

```
{{module "ngx_hello_module"}}
```

To load a built-in module like `ngx_stream_module`, add the following to the top of your `nginx.conf` file. You do not need to provide an `ngx_stream_module.so` file:

```
{{module "ngx_stream_module"}}
```

Note: To name your modules directory something other than `modules`, use the NGINX `load_module` directive, providing a path to the module relative to the location of your `nginx.conf` file. For example: `load_module some_module_dir/my_module.so`

Buildpack Support

The following resources can assist you when using the NGINX buildpack or when developing your own NGINX buildpack:

- **NGINX Buildpack Repository in GitHub**: Find more information about using and extending the NGINX buildpack in the NGINX buildpack GitHub repository.
- **Release Notes**: Find current information about this buildpack on the NGINX buildpack release page in GitHub.
- **Slack**: Join the #buildpacks channel in the Cloud Foundry Slack community.