How to choose the stack for your next website

Wednesday, September 21, 2022

Image taken from: https://michaelscodingspot.com/wp-content/uploads/2022/06/best-dotnet-web-application-tech-stack.jpg

Contents

1

Motivation

Motivation

Frequently as web developers, we are asked about implementing a new website for a client.

Most of the time, we tend to propose the same solution we have implemented before even if that solution is not the most optimal for the client's needs.

2

Common types of websites

Blogs – news

To educate or inform the visitor about current events or specialized knowledge.

What do we need?

Where to input content → Where to store content → Where to present content

Blogger | Wordpress | Wix | Google Sites | Medium

Ecommerce

To sell items online with a conventional retail method.

Where to input product content, check inventory, and manage orders → Where to store products → Where to present the products and handle users transactions

Wordpress

Shopify

Magento

Woocommerce

Business

To inform prospective clients and consumers about your business and entice them to work with you.

Where to input content → Where to store content → Where to present content

Wordpress

Wix

Portfolio

To display samples of work for certain professionals and attract more clientele.

Where to input content → Where to store content → Where to present content

Wordpress

Behance

Service provider

To offer a complete online service, such as streaming or online tools like search engines, spell-checkers, photo editors, or translators.

Where to input content → Where to store content → Where to present content

Lambda

Cloud Function

Landing page

To drive customers to a single, specific action, usually as part of a greater marketing campaign.

Where to input content → Where to store content → Where to present content

Wordpress

Wix

3

Stacks

Where to input content?

Disk	Content	Cloud database
JSON	Wordpress	Firabase
Ymal	Contentful	Amplify
Po files (translations)	Strapi	...
	Prismic	
	Wagtail	
	AEM	

Where to store content?

File databases

- SQLite

Search engines

- Elasticsearch
- OpenSearch
- Solr
- ArangoDB

Database servers

- MySQL
- Postgres
- MongoDb
- Cassandra

Cloud databases

- Firebase
- Dynamodb

Where to input product content, check inventory, and manage orders?

Disk

- JSON
- Ymal

Ecommerce

- Woocommerce
- Shopify
- PrestaShop
- Magento

Cloud databases

- Firebase
- Amplify

Where to present content?

Static pages	Client side rendering	Server side rendering
Pelican	Vanilla JS	Next
Lektor	React	Nuxt
MkDocs	Vue	Gastby
Grow SDK	Angular	Koa
Frozen-Flask	SolidJS	Express
	Svelte	Django
	Alpine.js	Flask
	Lit	

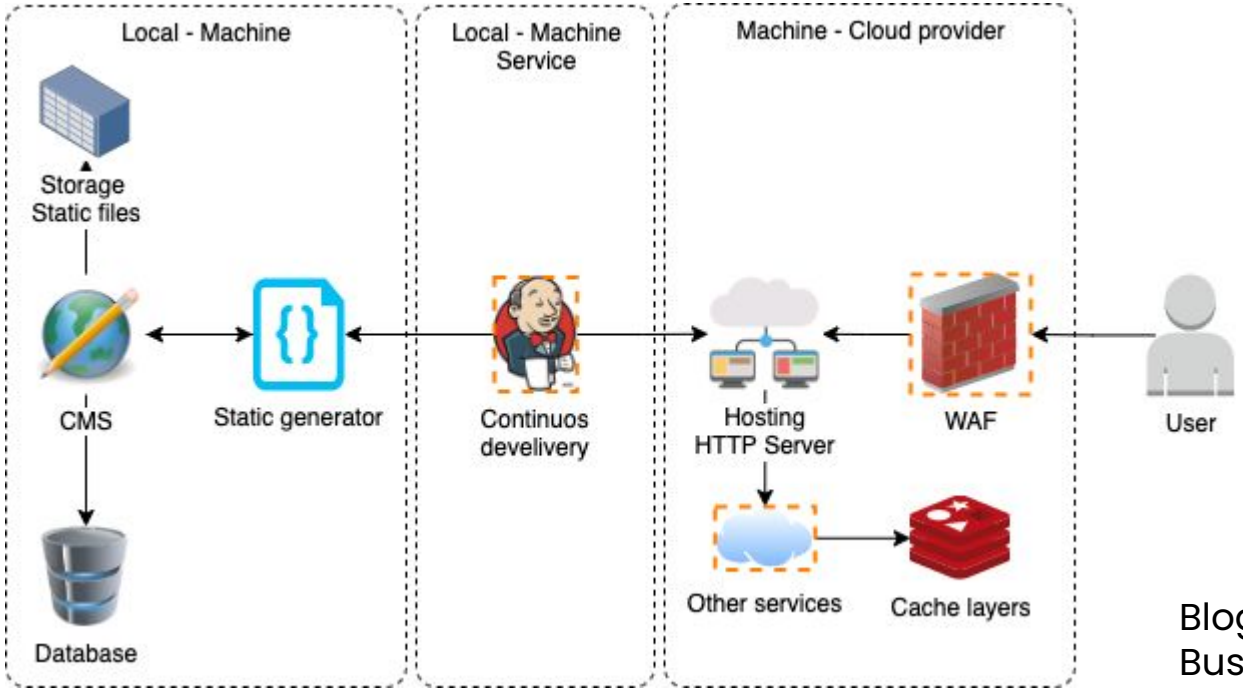
Where to present the products and handle users transactions?

Compiled HTTP Server

Client side rendering	Server side
Vanilla JS	Django
React	Flask
Vue	Express
Angular	NestJS
SolidJS	
Svelte	
Alpine.js	
Lit	

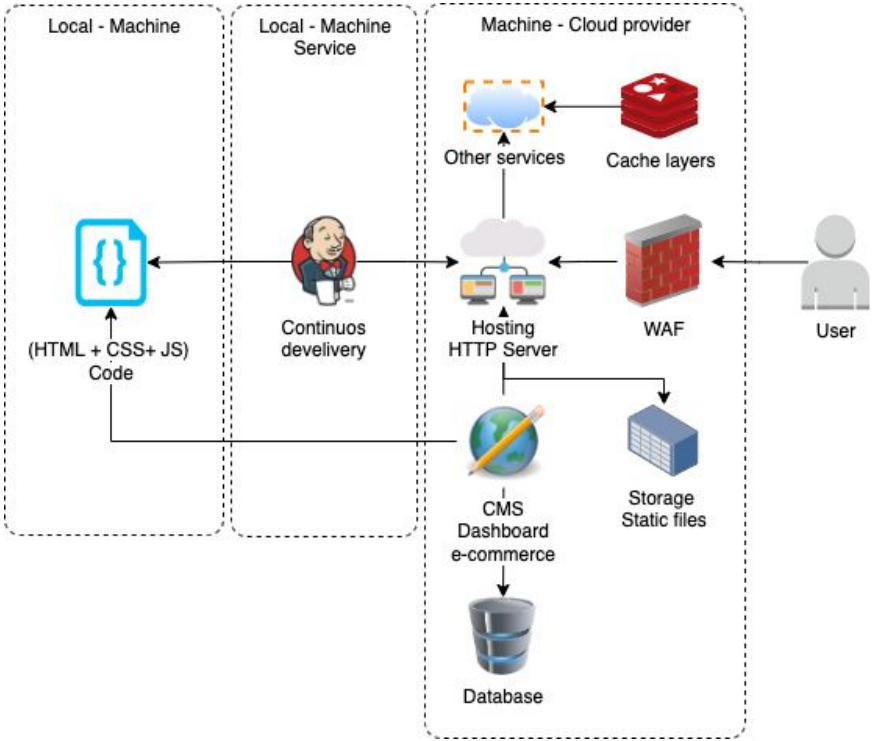
4

Architectures



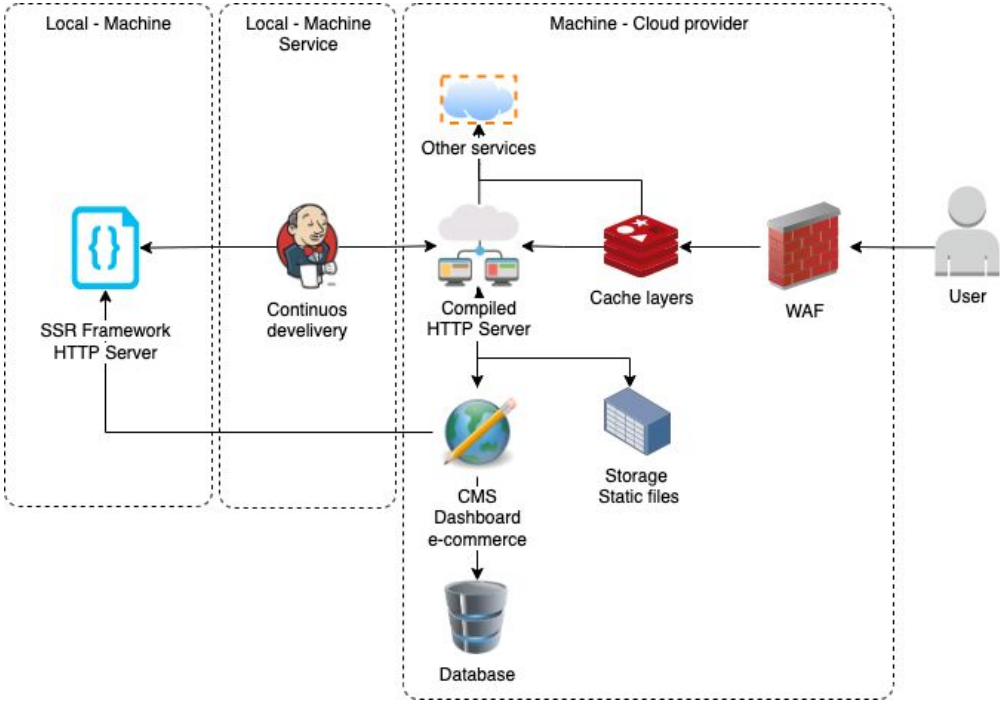
Blogs - news
Business
Portfolio
Landing page

Static website



Ecommerce
Dashboards
Admin sites

Dynamic websites



Service provider

Dynamic Content + SEO needs

5

Examples

Setup

Wagtail + Nuxt

Wagtail

1. **Create env:**

   ```
   python3 -m virtualenv venv
   ```

2. **Install Django and Wagtail**

   ```
   pip install Django==3.2.12 wagtail==2.15.5
   ```

3. **Create a wagtail project**

   ```
   wagtail start project
   ```

4. **Continue steps in:**
   https://wagtail.org/developers/

Set Wagtail API

1. **Add App:**

```
INSTALLED_APPS = [
    # API
    'rest_framework',
    'wagtail.api.v2',
]
```

2. **Add api.py config and api path is urls.py:**
   https://docs.wagtail.org/en/stable/advanced_topics/api/v2/configuration.html

Set Django Storages

1. **Install package for GCP:**

```
pip install "django-storages[google]"
```

2. **Set config in base.py:**

```
GS_CREDENTIALS = service_account.Credentials.from_service_account_file("credentials.json")

DEFAULT_FILE_STORAGE = 'storages.backends.gcloud.GoogleCloudStorage'

GS_BUCKET_NAME =

GS_QUERYSTRING_AUTH = False

GS_DEFAULT_ACL = 'publicRead'
```

Go to firebase and create a new project, a bucket in the storage section and download a service account key from project configurations.

Wagtail API

Custom Pages Api

OPTIONS GET

```
GET /api/v2/pages/3/
```

```
HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 3,
    "meta": {
        "type": "core.HomePage",
        "detail_url": "/api/v2/pages/3/",
        "html_url": "http://localhost:8000/",
        "slug": "python-medellin",
        "show_in_menus": false,
        "seo_title": "",
        "search_description": "",
        "first_published_at": "2022-09-21T20:07:36.795872Z",
        "parent": null,
        "relative_url": "/"
    },
    "title": "Python Medellin",
    "main_header": "Meetup Python Medellin",
    "hero_image": {
        "xs": {
            "url": "https://storage.googleapis.com/python-meetup-363200.appspot.com/images/clean_505480862.width-375.jpg",
            "width": 375,
            "height": 210,
            "alt": "Python Medellin"
        },
```

Nuxt

1. **Use nvm or something to lock Node version:**

   `lts/gallium`

2. **Create a Nuxt App:**

   `yarn create nuxt-app frontend`

   Recommended options:

   JavaScript, Yarn, UI framework: None, Axios, All Linting tools, Jest.

   Rendering mode: Universal

   Deployment target: Static (static, dynamic) / Server(SSR)

   Continuous integration: None

   Git

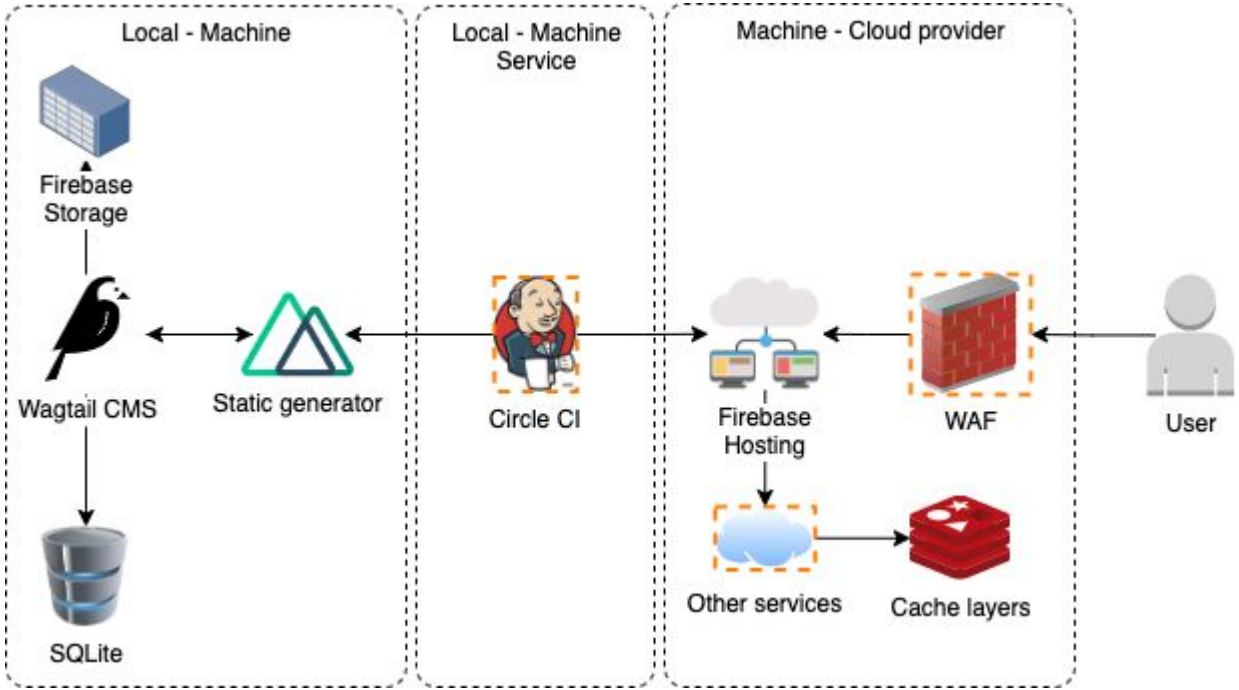
Static Website

Settings in nuxt.config.js:

```
target: 'static',
```

Repo:

Talks - 22-09-21 - Static Website



Static website

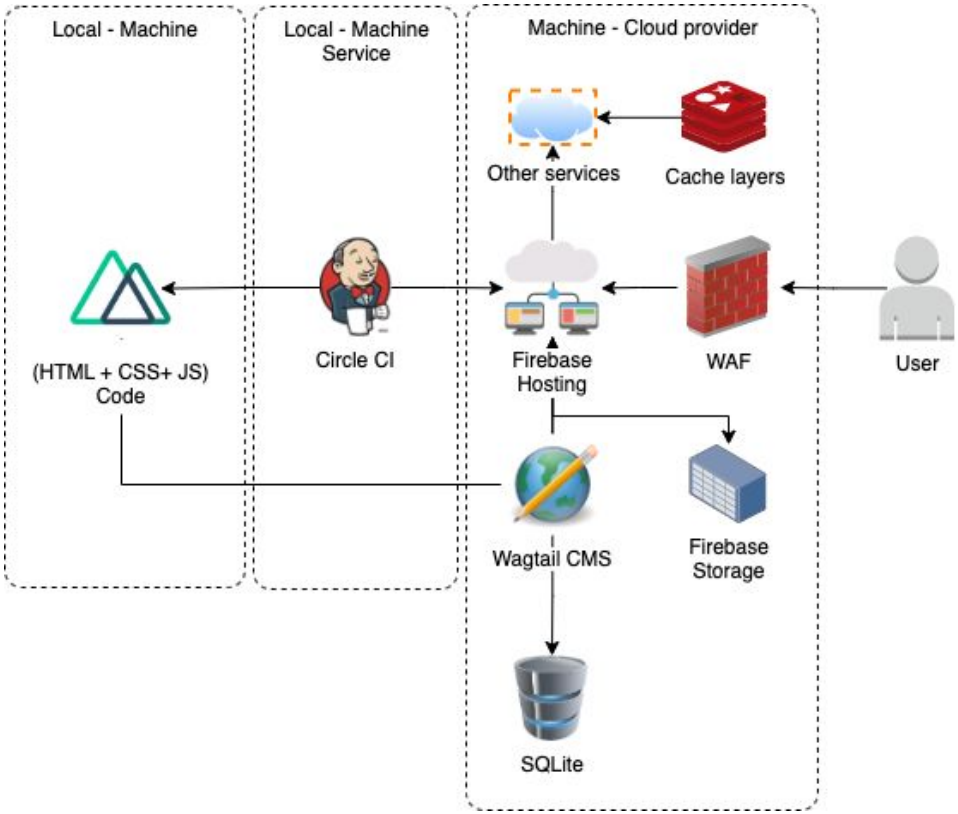
Dynamic Website

Settings in nuxt.config.js:

```
ssr: false,
target: 'static',
```

Repo:

[Talks - 22-09-21 - Dynamic Website]



Dynamic websites

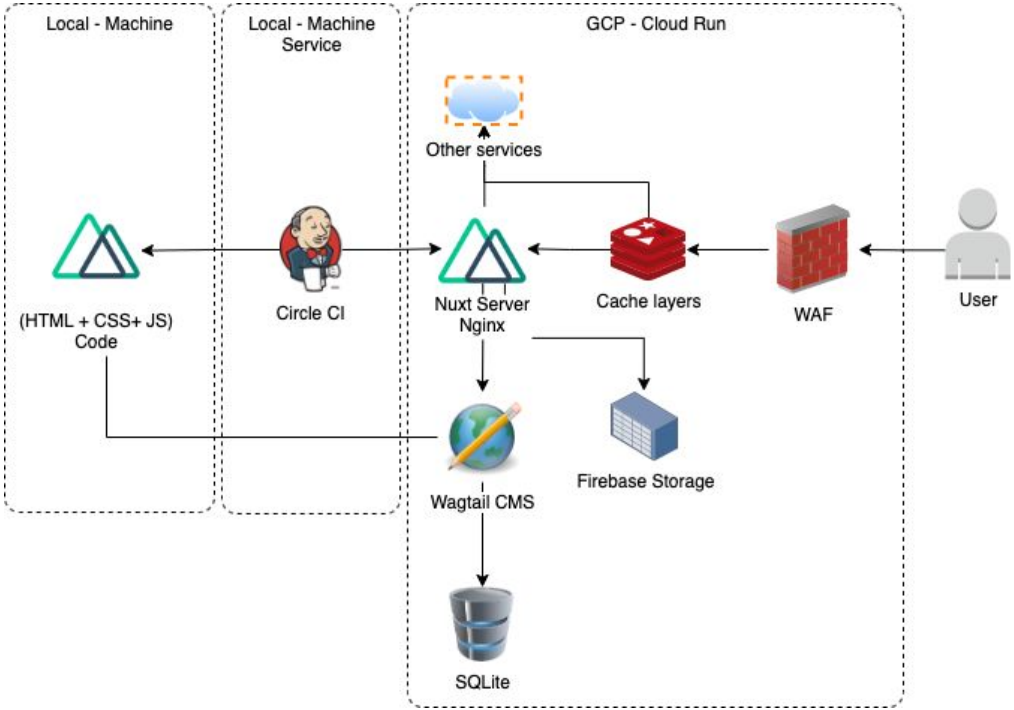
SSR Website

Settings in nuxt.config.js:

```
ssr: true,
target: 'server',
```

Repo:

Talks - 22-09-21 - Server Side Website



Dynamic Content + SEO needs