



EUROPE'S LEADING AEM DEVELOPER CONFERENCE
28th – 30th SEPTEMBER 2020

Introducing Kestros
Danny Moulton, Kevin O'Brien, Kestros Inc.

In This Talk

- Development Focus
- UI Frameworks & Components
- Administrative Features
- Auto-Generated Proxy Configuration
- Containerization

Development Focus

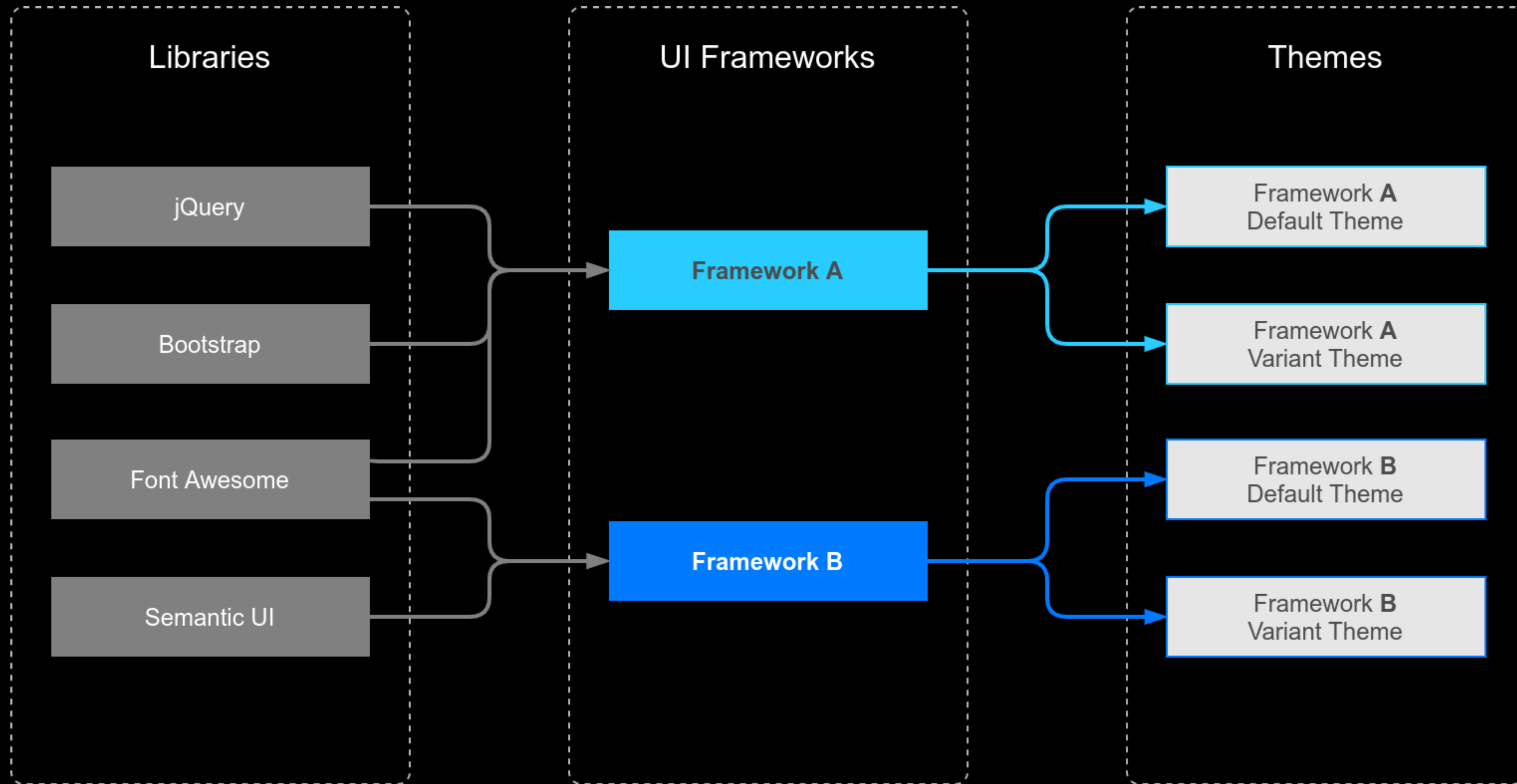
Development Focuses

- Developers and designers first
- Easy for newcomers, powerful for experts
- Platform as a companion during development
- Meaningful feedback
- Maximize reusability
- Unopinionated but encourages best practices

UI Frameworks and Components Types

- **Modular Frontend Libraries**
 - Contains CSS/JS
 - Bundled HTL Templates
- **UI Frameworks**
 - Compilation of frontend libraries
 - Themes
 - Implemented by sites/pages

UI Frameworks & Components Types



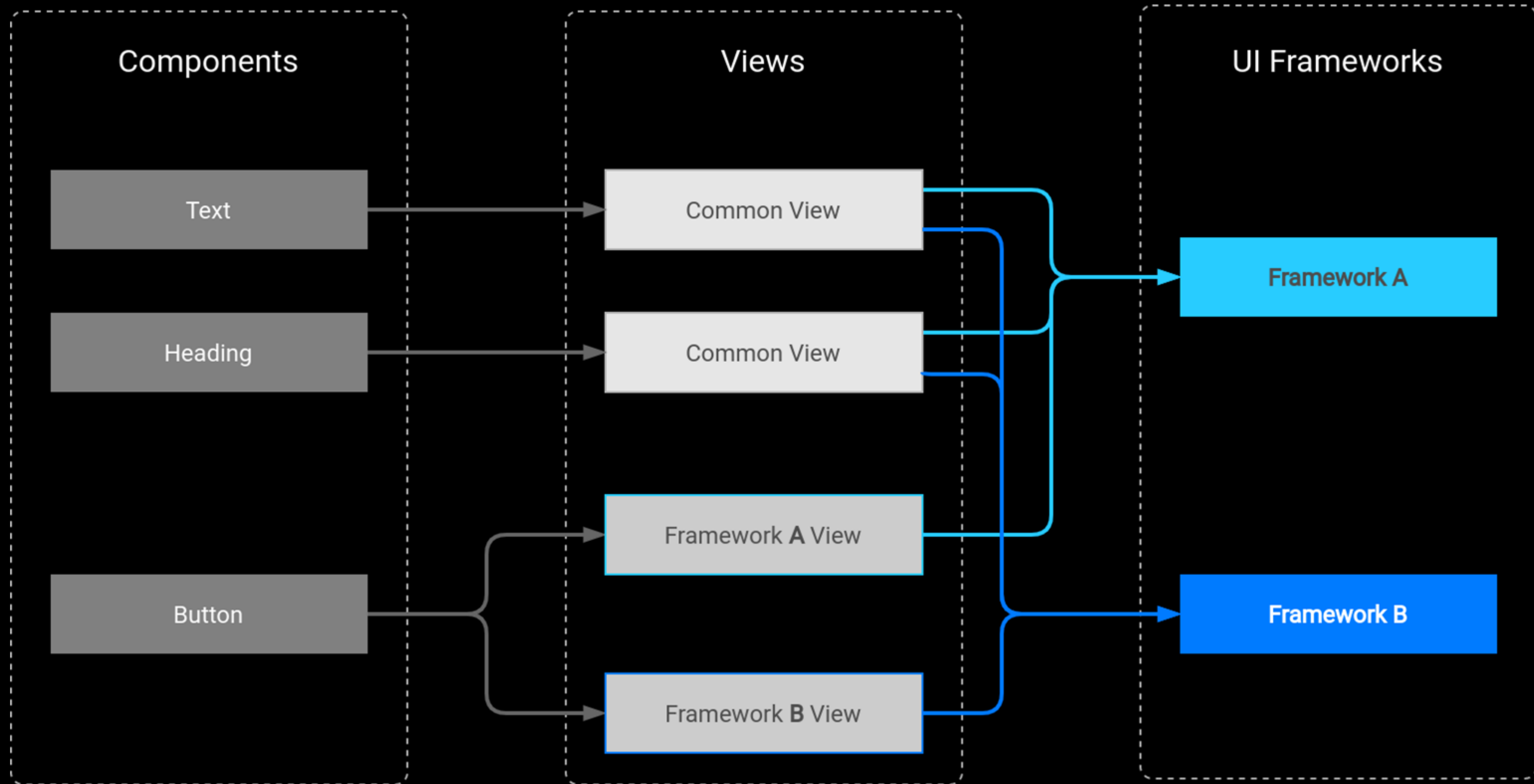
Component Types

- Sling Model
- Dialog Configurations (create, edit, delete)
- UI Framework Views
 - Modular Views per UI Framework
 - Common View (to gracefully fall back to)
- Models and Dialogs - code once, reuse across all views

Component UI Framework Views

- Maximize HTL Template reuse
- Modular variations
- Switch between UI Frameworks seamlessly

UI Frameworks & Components Types



Administrative Features

Administrative Features

- Everything can be validated
- Automated documentation
- Automated troubleshooting
- Deployable with or without administrative features

SlingModelUtils

SlingModelUtils

- **SlingModelUtils.adaptTo()**

```
try {  
    return SlingModelUtils.adaptTo(resource, BaseComponent.class);  
} catch (InvalidResourceTypeException e) {  
    // Do something or log.  
}
```

- **Other SlingModelUtils Methods**

- **getChildrenOfType()**
- **getResourcesAsType()**
- **getAllDescendantsOfType()**
- **getResourceAsClosestType()**
- **Many more...**

Auto-Generated Proxy Configuration

Auto-Generated Proxy Configuration

- No more manual filter configuration
- Ships with NGINX as the default proxy
- Apache HTTPD support in development
- Configuration for resources is generated during publish
- Location directive configuration blocks
 - proxy_pass settings
 - proxy_cache settings
 - sub_filter settings

- **Sample location configuration block**

```
location /getting-started.html {  
    sub_filter_once off;  
    sub_filter_types text/html;  
    sub_filter "public/sites/kestros-io/" "";  
    proxy_pass http://kestros-app-pool/public/sites/kestros/getting-started.html;  
    proxy_redirect off;  
    proxy_set_header Host $host;  
    add_header Cache-Control "public, max-age=86400";  
    proxy_cache kestros;  
    proxy_cache_revalidate on;  
    proxy_cache_use_stale error timeout updating http_500 http_502 http_503 http_504;  
    proxy_cache_background_update on;  
    proxy_cache_lock on;  
}
```

Auto-Generated Proxy Configuration

- Location block assembled from models for each type of config

```
StringJoiner cacheConfig = new StringJoiner(System.getProperty("line.separator"));
cacheConfig.add("    proxy_cache " + this.proxyCacheName + ";");
...
this.cacheControlHeaders.forEach((headerKey, headerValue) -> {
    cacheConfig.add("    add_header " + headerKey + " " + headerValue);
});
return cacheConfig.toString();
```

Auto-Generated Proxy Configuration

- Finding resources to generate configuration for

```
package io.kestros.proxy.nginx;

import io.kestros.cms.foundation.content.pages.BaseContentPage;
import io.kestros.cms.foundation.content.sites.BaseSite;
import io.kestros.commons.structuredslingmodels.utils.SlingModelUtils;
...
BaseSite site = SlingModelUtils.adaptTo(request.getResource(), BaseSite.class);
...
for (BaseContentPage page : SlingModelUtils.getAllDescendantsOfType(site,
    BaseContentPage.class)) {
    ...
    locationConfigFileData.addAll(Arrays.asList(locationConfig.toString()));
}
```

Auto-Generated Proxy Configuration

- Shared volume mounted on Kestros & NGINX containers
- Site resource tree scanned for relevant resource types
 - Configured in `@KestrosModel` annotation on Sling Models
- Location block generated for each resource
- Configuration file written by Kestros container

Containerization

Containerization

- Containerized & designed to be Cloud-Native
- Externalized configuration: build once, run everywhere
- Custom run modes enabled via provisioning model

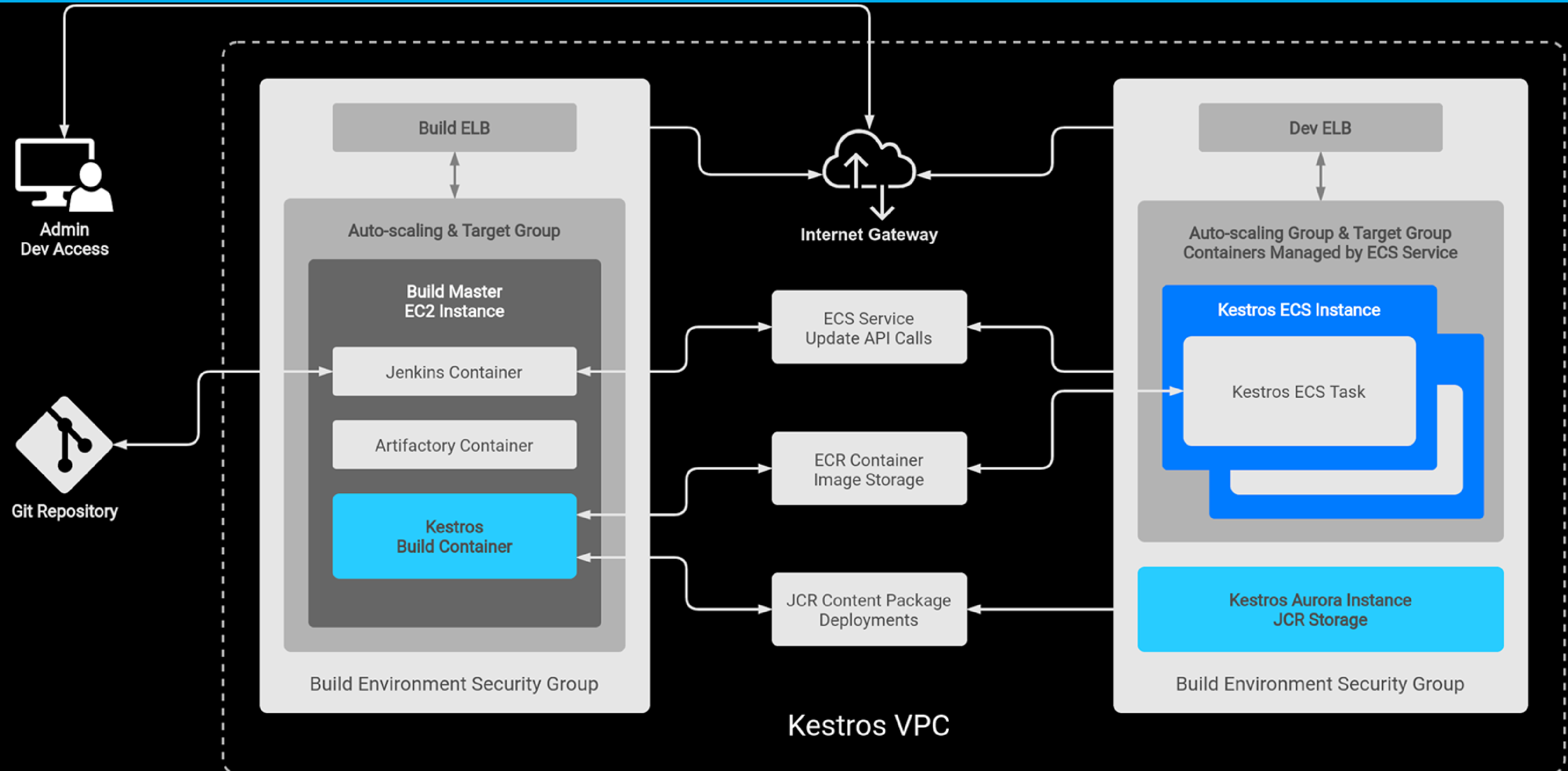
```
-Dsling.run.modes=oak_mysql ... -  
Dsling.fileinstall.dir=/opt/sling/ext-config
```

```
/keystros/config/org.apache.sling.datasource.DataSourceFactory-oak.config -> /opt/sling/ext-  
config/
```

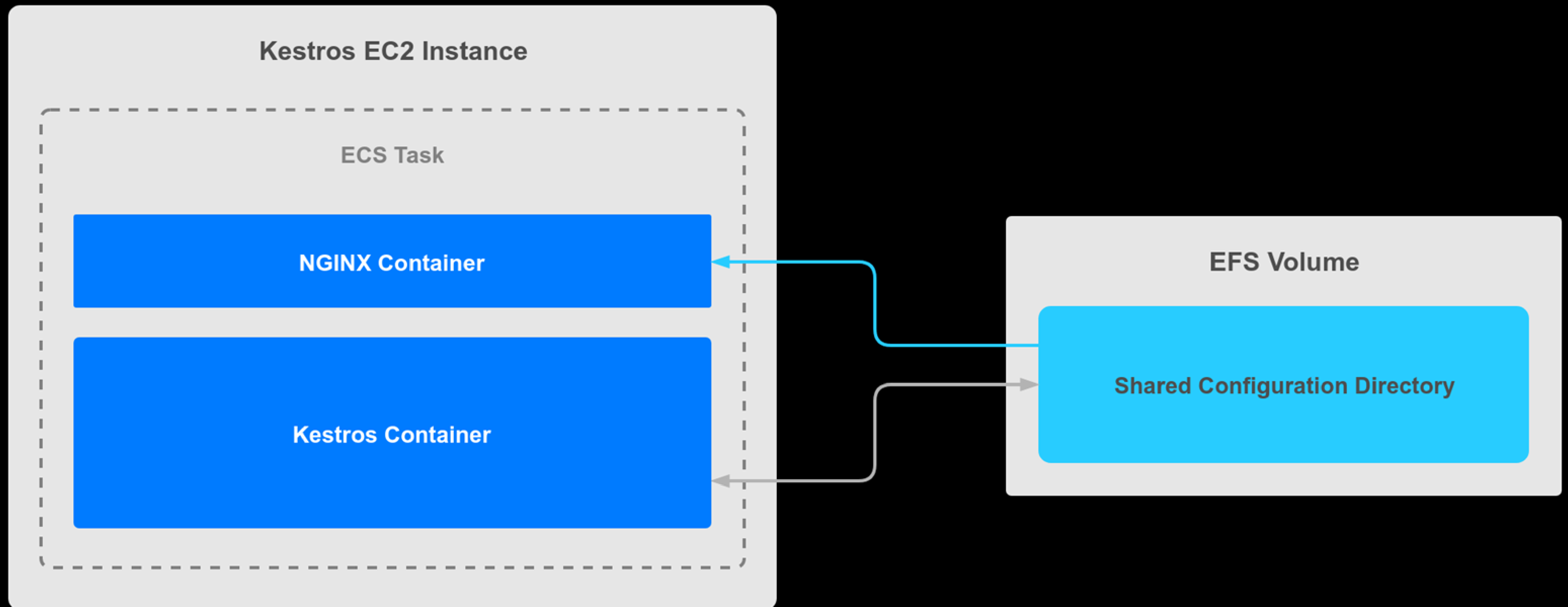
Containerization

- Docker compose files for local development
- ECS task definitions / CloudFormation templates for AWS
- Kubernetes pod configuration for everywhere else

AWS Development Environment



Kestros Instance - ECS Example



What's Next?

What's Next

- Try the Kestros Alpha
<https://kestros.io/getting-started>
- Follow the development
<https://github.com/kestros>
<https://kestros.io>
- Connect
info@kestros.io
[@KestrosInc](#)