

# Application Manifests

# Topics

- **Introduction to Application Manifests**
- Application Manifest Features

# Application Manifest



- A client-side file that describes the application deployment options
  - Provides repeatability when pushing applications
  - Same options as `cf push` command
- Default name: `manifest.yml`
  - YAML format
  - Human friendly data serialization standard
  - Supported by many programming languages
  - Less verbose than XML, similar to JSON
  - <http://www.yaml.org>

# Using an Application Manifest with cf push



- `cf push` automatically detects the application manifest in the current directory
  - Expects a file named `manifest.yml`
- Can override the manifest name or location with `-f` option
  - `cf push -f dev-manifest.yml`
- Ignore existing manifests with `--no-manifest` option
- No manifest found?
  - `cf push` will use command line arguments
  - Default values for other deployment options

<http://docs.pivotal.io/pivotalcf/devguide/deploy-apps/manifest.html>

# YAML Format



- Three dashes
  - Indicate start of document
- Indent with spaces, not tabs!
  - Determines hierarchy
  - Each indent 2 spaces
  - “-” defines “group”
- Syntax: `property: value` pairs
- `#` starts a one line comment

```
---
applications:
- name: nodetestdh01
  memory: 64M
  instances: 2
  host: crn      # comment
  domain: cfapps.io
  # comment
- name: nextapp      # group 2
  memory: 256M
  ...
```

See <http://www.yaml.org/spec/1.2/spec.html>

# manifest.yml Example

- applications: can describe one or more applications
- name: of the app – used in commands
- command: command used to start the app (optional)
- memory: app memory limit (1 GB default)
- instances: app instances
- host: subdomain of the URL
- domain: last part of the URL

```
---
applications:
- name: cf-node-demo
  command: node app.js
  memory: 128M
  instances: 1
  host: myapp
  domain: cfapps.io
```

# Topics

- Introduction to Application Manifests
- **Application Manifest Features**

# cf push Parameter Precedence

- 1) Command line arguments (always take precedence)
- 2) Application manifest values (if not in the command line)
- 3) The last value the server used
- 4) Default server values

```
cf push -m 256M  
cf push -i 2
```

```
---  
applications:  
- name: myapp  
  host: myappa  
  domain: cfapps.io  
  instances: 3
```

Question: If an app has been pushed two times using the commands and manifest above, what will be the resulting:

- a) application name
- b) URL to the application
- c) number of instances
- d) application memory



# Why Include Multiple Applications?



- If you specify the application name in the command line
  - `cf push <appname>`
- The parameters from the application manifest for that name will be used
- Only one application will be pushed
- Useful for testing different manifest settings
- If you don't pass a name in the command line, all apps are pushed (must specify paths)

```
---
applications:
- name: applowmemory
  memory: 64M
  host: crn-low
  path: lowmem/applow.jar

- name: apphighmemory
  memory: 256M
  host: crn-high
  path: highmem/apphigh.jar
...
```

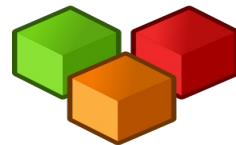
# Specifying Timeouts for Large Apps



- Can use the manifest to specify timeouts for starting, staging or startup
  - timeout: default is 60 seconds for startup, maximum is 180 seconds
  - Use environment variables for server staging and startup timeouts (in minutes)

```
---
applications:
- name: nodetestdh01
  memory: 64M
  instances: 2
  host: crn
  domain: cfapps.io
  timeout: 120 # 120 secs
  env:
    # Set to 20 and 10 mins
    CF_STAGING_TIMEOUT: 20
    CF_STARTUP_TIMEOUT: 10
```

# Note on Spaces



- A space cannot be specified in a manifest file
  - Set first: `cf target -s development`
- To determine the current space
  - Just run: `cf target`

# Topics

- Introduction to Application Manifests
- Application Manifest Features

```
---  
applications:  
- name: ???  
  memory: 128M  
  instances: 1
```

## Lab

Use an application manifest with your application