Google Summer of Code

- https://jenkins.io/projects/gsoc/students/
Triggering builds behind firewall

- Services cannot reach internal Jenkins
- Public Jenkins is a security thread / operation challenge
- Pains of polling
- [https://jenkins.io/blog/2019/01/07/webhook-firewalls/](https://jenkins.io/blog/2019/01/07/webhook-firewalls/)
Touchless Jenkins

- It does not end with pipeline
- Job DSL plugin makes it to the spotlight
- Entering Jenkins Configuration as Code
Things that pipeline does not cover

- Job creation itself
- Structure: folders/views
How can Job DSL help with that?

- Creating folders and views
- Defining job steps obsoleted by Pipeline
- Traditionally invoked as Jenkins (seed) job
  - Chicken & Egg
What is JCasC?

- YAML definition of global configuration
  - Authentication, Nodes, Credentials, etc.
- Injected during startup
- Reapplied without restart
- Under active development
## Going touchless

<table>
<thead>
<tr>
<th>Feature</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Configuration</td>
<td>JCasC (YAML)</td>
</tr>
<tr>
<td>Folders, Views, Jobs</td>
<td>Job DSL (Groovy)</td>
</tr>
<tr>
<td>Job Definition</td>
<td>Pipeline (Groovy)</td>
</tr>
</tbody>
</table>
JCasC invocation

- Required plugins
  - configuration-as-code, configuration-as-code-support
- Config Injected through variable CASC_JENKINS_CONFIG
  - From file(s) on Jenkins master
  - From remote URL
JCasC definition essentials

- JENKINS_URL/configuration-as-code/reference
- Examples

```yaml
jenkins:
  ...
  nodes:
    ...
  clouds:
    ...
  jobs:
    ...
  unclassified:
    ...
  tool:
    ...
```
Job DSL

- https://jenkinsci.github.io/job-dsl-plugin/

```java
folder('foo') {
    views {
        listView('bees') {
            jobs { regex(/b.*/) }
            columns { name(); lastSuccess() }
        }
    }

    job('foo/bar') { ... }
}
```
How that all fit together?

jenkins:
  numExecutors: 1

jobs:
  - script: >
    pipelineJob('foo') {
      definition {
        cps {
          script(""
            node() {
              sh ""
              hostname
            }
            ""
          )
          sandbox()
        }
      }
    }
Sustainable config

- Separate dialects
  - Invoke Job DSL through file or url specifier
  - Refer to pipeline scripts in SCM repositories
Sustainable config

CASC_JENKINS_CONFIG=https://raw.githubusercontent.com

jobs:
  - file: job-dsl.groovy
  - url: ...

pipelineJob('example') {
  definition {
    cpsScm {
      scm {
        git('https://github.com/foo/bar.git')
      }
    }
  }
}
Discussion