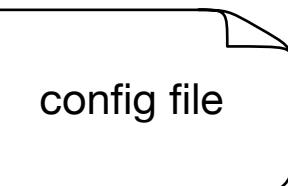


CONSUL  
TEMPLATE



config file



BOM (Bill Of Materials)

Helm Chart



Helm



us-west-1

Physical or Virtual Data Center



us-east-2

Physical or Virtual Data Center



Kubernetes



QA Cluster

Kubernetes



Prod Cluster

Kubernetes



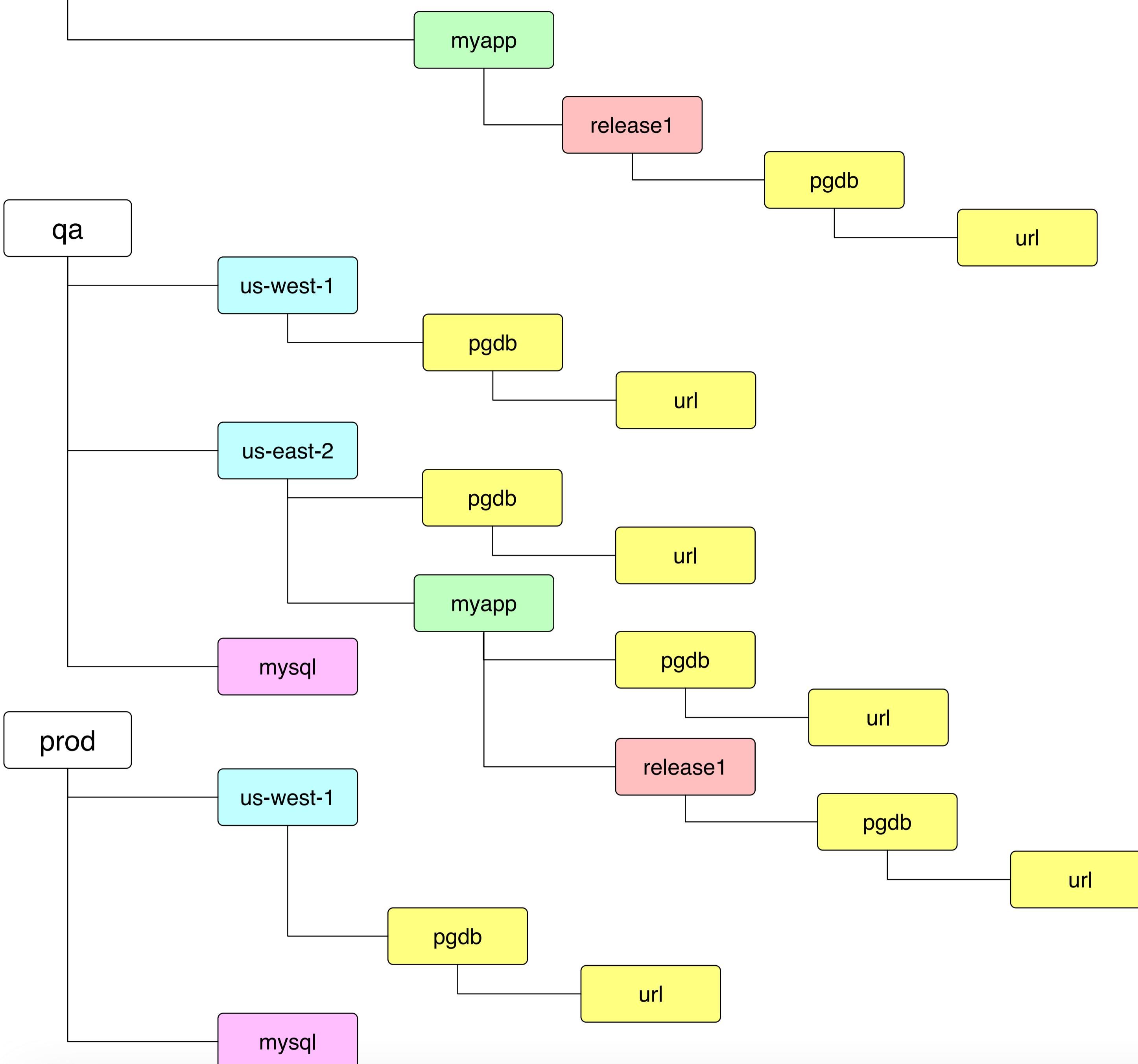
Pre-prod Cluster

Kubernetes



Prod Cluster

default



```
3  consul kv put default/count 1
4  consul kv put default/pgdb/url "postgresql://localhost/db"
5  consul kv put qa/us-west-1/pgdb/url "postgresql://us-west-1/db-qa"
6  consul kv put prod/us-west-1/pgdb/url "postgresql://us-west-1/db-prod"
7  consul kv put us-east-2/pgdb/url "postgresql://east/db"
8  consul kv put qa/us-east-2/myapp/pgdb/url "postgresql://east-myapp/db"
9  consul kv put qa/us-east-2/myapp/release1/pgdb/url "postgresql://east-myapp/db-r1"
10 consul kv put myapp/pgdb/url "postgresql://localhost/mydb"
11 consul kv put default/myapp/release1/pgdb/url "postgresql://localhost/mydb-r1"
12 consul kv put qa/mysql @mysql1.json
13 consul kv put prod/mysql @mysql2.json
```

```
2 got count [{{ key "default/count" }}], expected <1> - direct key lookup
3 {{ $location := or (env "CLUSTER_LOCATION" ) "us-west-1" }}
4 {{ $app := or (env "APP" ) "myapp"}}
5 {{ $release := or (env "RELEASE" ) "release1"}}
6 {{ $env := or (env "ENVIRONMENT" ) "qa"}}
7
8 {{$location}}_{{$env}}={{plugin "hiera-ctp" "pgdb/url" $location $env}}
9
10 {{$app}}_{{$location}}_{{$env }}={{plugin "hiera-ctp" "pgdb/url" $app $location $env}}
11
12 {{$release}}_{{$app}}_{{$location}}_{{$env }}={{plugin "hiera-ctp" "pgdb/url" $release $app $location $env}}
13
14 Getting JSON value
15
16 {{with $d := plugin "hiera-ctp" "mysql" $release $app $location $env | parseJSON}}
17 url = {{$d.url}}
18 user = {{$d.user}}
19 {{end}}
```

```
19 export CLUSTER_LOCATION=us-west-1
20 export RELEASE=release1
21 runTemplates
22 export CLUSTER_LOCATION=us-east-2
23 runTemplates
24 export RELEASE=release2
25 runTemplates
26 export RELEASE=release2
27 runTemplates
28 export ENVIRONMENT=prod
29 runTemplates
```

got count [1], expected <1> - direct key lookup [1], expected <1> - direct key lookup

|  |  |  |
|--|--|--|
| us-east-2_qa=postgresql://localhost/db                 | 6<br>7   | 6<br>7   |
| myapp_us-east-2_qa=postgresql://east-myapp/db          | 8<br>9   | 8<br>9   |
| release2_myapp_us-east-2_qa=postgresql://east-myapp/db | 10<br>11   | 10<br>11   |
| Getting JSON value                                     | 12<br>13<br>14<br>15<br>16<br>17<br>18<br>19<br>20 | 12<br>13<br>14<br>15<br>16<br>17<br>18<br>19<br>20   |
| url = jdbc:mysql://mysql1:3306/users<br>user = user1   | 17<br>18<br>19<br>20                               | url = jdbc:mysql://mysql1:3306/users<br>user = user1 |

```
#!/usr/bin/env bash

set -e

mvn install
cat stub.sh target/hiera-ctp-0.1-SNAPSHOT-spring-boot.jar > target/hiera-ctp
chmod u+x target/hiera-ctp
```

```
1 #!/bin/sh
2 https://coderwall.com/p/ssuaxa/how-to-make-a-jar-file-linux-executable
3 MYSELF=`which "$0" 2>/dev/null`
4 [ $? -gt 0 -a -f "$0" ] && MYSELF="./$0"
5 java=java
6 if test -n "$JAVA_HOME"; then
7     java="$JAVA_HOME/bin/java"
8 fi
9 exec "$java" $java_args -jar $MYSELF "$@"
10 exit 1
```

```
Running hiera-ctp-java/target/hiera-ctp
```

```
real    0m3.156s  
user    0m5.264s  
sys     0m0.493s
```

```
Running /Users/kgignatyev/.cargo/bin/hiera-ctp
```

```
real    0m0.130s  
user    0m0.060s  
sys     0m0.048s
```

```
Running /Users/kgignatyev/go/bin/hiera-ctp
```

```
real    0m0.125s  
user    0m0.061s  
sys     0m0.046s
```

```
congratulations, rendered templates are the same  
congratulations, rendered go templates are the same  
Running hiera-ctp-java/target/hiera-ctp
```

```
real    0m3.231s  
user    0m5.380s  
sys     0m0.518s
```

```
Running /Users/kgignatyev/.cargo/bin/hiera-ctp
```

```
real    0m0.140s  
user    0m0.063s  
sys     0m0.052s
```

```
Running /Users/kgignatyev/go/bin/hiera-ctp
```

```
real    0m0.128s  
user    0m0.062s  
sys     0m0.047s
```

```
..$ ls -alh hiera-ctp-java/target/hiera-ctp  
 3.2M May 15 14:37 hiera-ctp-java/target/hiera-ctp  
ev$ ls -alh ~/.cargo/bin/hiera-ctp  
 4.9M May 15 14:36 /Users/kgignatyev/.cargo/bin/hiera-ctp  
ev$ ls -alh ~/go/bin/hiera-ctp  
 6.9M May 15 14:37 /Users/kgignatyev/go/bin/hiera-ctp
```

```

58 ► fn main() {
59     let args: Vec<String> = env::args().collect();
60     let (left: &[String], path_parts: &[String]) = args.split_at(mid: 2);
61     let key: &String = left.get(index: 1).unwrap();
62     let mut res: Option<String> = find_key_value(&key, &path_parts);
63     match res {
64         Some(data: String) => {
65             res = Some(data);
66         }
67         None => {
68             let default_path: String = format!("default/{}", key);
69             let default_res: Result<String, CustomError> = get_key_value(&default_path);
70             res = match default_res {
71                 Ok(s: String) => Some(s),
72                 Err(_) => None
73             }
74         }
75     }
76
77     match res {
78         Some(v: String) => println!("{}: {}", v),
79         None => println!("null")
80     }
}

```



#Rustacean



```

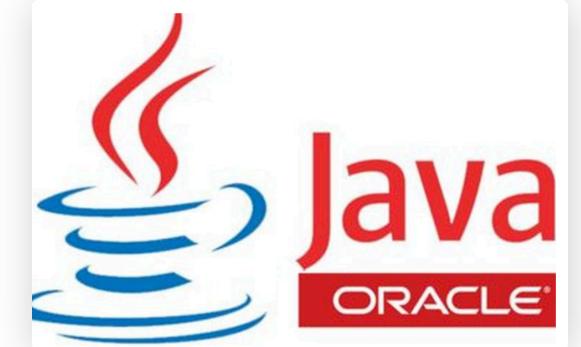
13 fn get_key_value(path: &String) -> Result<String, CustomError> {
14     let url: String = format!("http://localhost:8500/v1/kv/{}", path);
15     //    println!("trying:{}", url);
16
17     let mut res: Response = reqwest::get(&url)?;
18     let mut body: String = String::new();
19     res.read_to_string(buf: &mut body)?;
20     if res.status() == 200 {
21         let r: Result<Value, serde_json::Error> = serde_json::from_str(s: &body);
22         match r {
23             Ok(v: Value) => {
24                 let b64v: &str = v[0]["Value"].as_str().unwrap();
25                 let vd: Vec<u8> = decode(input: b64v).unwrap();
26                 let s: String = String::from_utf8(vec: vd).unwrap();
27                 Ok(s)
28             }
29             Err(_) => Err(CustomError::Info("no value".to_string()))
30         }
31     } else {
32         Err(CustomError::Info("no value".to_string()))
33     }
34 }

```

```

74 @ ...
75     public static Result find_key_value(String key, String[] path_parts) {
76         var parts = Arrays.asList(path_parts);
77         Collections.reverse(parts);
78         String res = null;
79         for (int x = 0; x < parts.size(); x++) {
80             if (res == null) {
81                 StringBuilder sb = new StringBuilder();
82                 for (int i = 0; i < parts.size() - x; i++) {
83                     if (sb.length() != 0) {
84                         sb.append("/");
85                     }
86                     sb.append(parts.get(i));
87                 }
88                 sb.append("/").append(key);
89                 String path = sb.toString();
90                 res = get_key_value_or_null(path);
91             }
92         }
93         if (res == null) {
94             // lets try to find a default
95             res = get_key_value_or_null(path: "default/" + key);
96         }
97         return new Result(res, responseCode: 200);
98     }
99
100    public static void main(String[] args) {
101        var path = Arrays.copyOfRange(args, from: 1, args.length);
102        var r = find_key_value(args[0], path);
103        var p = String.join(delimiter: ",", args);
104        System.out.println(r.res);

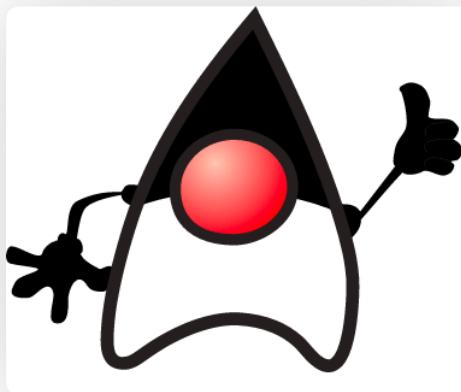
```



```

100    public static void main(String[] args) {
101        var path = Arrays.copyOfRange(args, from: 1, args.length);
102        var r = find_key_value(args[0], path);
103        var p = String.join(delimiter: ",", args);
104        System.out.println(r.res);

```



```

56 // this works faster than method with fluent HTTP that tests for return code before at
57 @ ...
58     public static Result get_key_value(String path) {
59         try {
60             var tree = om.readTree(new URL(spec: "http://localhost:8500/v1/kv/" + path));
61             var b64v = tree.get(0).get("Value").asText();
62             String val = new String(Base64.getDecoder().decode(b64v));
63             return new Result(val, responseCode: 200);
64         } catch (IOException e) {
65             // e.printStackTrace();
66         }
67         return new Result(res: "42", responseCode: 1);
68     }

```

Really bad piece of code!

```

29  func get_key_value(key string, path []string) (string, error) {
30      prefixPath := strings.Join(reverse(path), sep: "/")
31      resp, err := http.Get("http://localhost:8500/v1/kv/" + prefixPath + "/" + key)
32      var js []KeyData
33      if err == nil {
34          defer resp.Body.Close()
35          if resp.StatusCode == http.StatusOK {
36              bodyBytes, err := ioutil.ReadAll(resp.Body)
37              if err == nil {
38                  unmarshalErr := json.Unmarshal(bodyBytes, &js)
39                  if unmarshalErr != nil {
40                      return "", unmarshalErr
41                  }
42                  decoded, err := base64.StdEncoding.DecodeString(js[0].Value)
43                  return string(decoded), err
44              }else{ return "", err }
45          }else{
46              return "", errors.New( fmt.Sprintf( format: "status code was %v ", resp.StatusCode))
47          }
48      }
49      return "", err
50  }
51  func find_key_value(key string, path []string) (string,error) {
52      idx := 0
53      var v = ""
54      var err error
55      for idx < len(path){           |
56          p := path[idx:]
57          v,err =get_key_value( key, p)
58          if err == nil {
59              return v,nil
60          }
61          idx = idx+1
62      }
63      return v, errors.New( text: "not found")
64  }
65  func main() {
66      key,path := os.Args[1],os.Args[2:]
67      v,err:=find_key_value(key,path)
68      if err!= nil {
69          v,err = find_key_value(key, []string{"default"})
70      }
71      if err !=nil {
72          v = "null"
73      }
74      fmt.Println(v)
75      //println(v)    //prints to stdERROR

```

# GOLANG

