



Backstage

Presenting by Le Canh Duy

BackStage

template

What Backstage template can do ?

Help developers to create components inside Backstage

Ability to load **skeletons of code**

Publish the template to some location - Github, Gitlab



One structure for all ?

The screenshot shows the 'Create a new component' page in the Backstage framework. It features a dark green header with the title 'Create a new component' and a subtitle 'Create new software components using standard templates'. Below the header, there's a section titled 'Available templates'. On the left, a sidebar contains icons for various Backstage features. The main content area displays two template cards: 'React SSR Template' (labeled 'website') and 'Spring Boot Service' (labeled 'service'). Each card has a 'Recommended' tag, a language tag ('React' or 'Java'), a description, and a 'CHOOSE' button. A note at the top of the templates section states: 'NOTE! This feature is WIP. You can follow progress [here](#).'

Create a new component *alpha*

Create new software components using standard templates

Available templates

NOTE! This feature is WIP. You can follow progress [here](#).

website

React SSR Template

Recommended React

Next.js application skeleton for creating isomorphic web applications.

CHOOSE

service

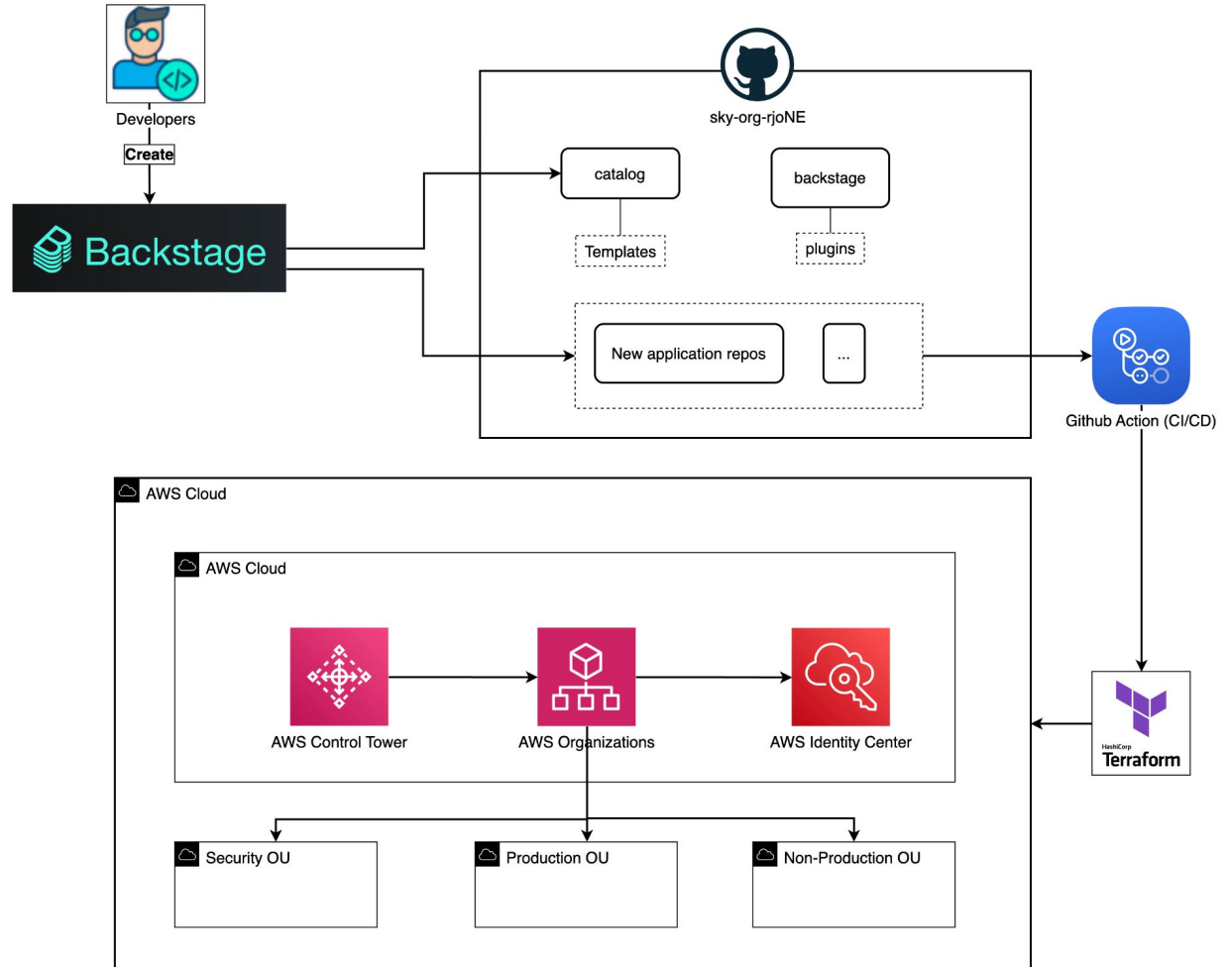
Spring Boot Service

Recommended Java

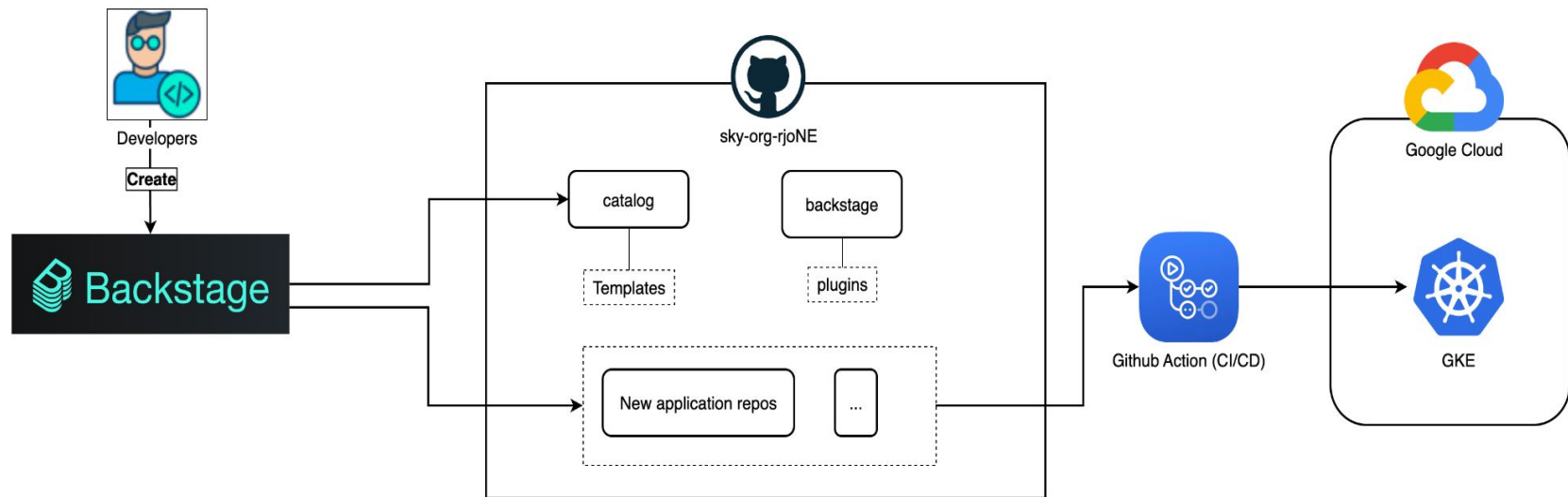
Standard Spring Boot (Java) microservice with recommended configuration.

CHOOSE

Terraform template



My BackStage instance



backstage/app-config.yaml

Github Integration

```
6 integrations:
7   github:
8     - host: github.com
9     apps:
10      - $include: github-app-backstage-5v5kwlxqwlgmuityej-credentials.yaml
11
12   auth:
13     environment: development
14     providers:
15       github:
16         development:
17           $include: credentials.yaml
```

backstage/app-config.yaml

Kubernetes Integration

```
101 kubernetes:
102   serviceLocatorMethod:
103     type: 'multiTenant'
104   clusterLocatorMethods:
105     - type: config
106       clusters:
107         - url: ${GKE_URL}
108           name: allnow-tms-dev-cluster
109           authProvider: serviceAccount
110           skipTLSVerify: true
111           skipMetricsLookup: false
112           serviceAccountToken: ${GKE_TOKEN}
113
```


backstage/catalog-info.yaml

Templates

```
84 catalog:
85   import:
86     entityFilename: catalog-info.yaml
87     pullRequestBranchName: backstage-integration
88   rules:
89     - allow: [Component, System, API, Resource, Location, Template, User, Group]
90   locations:
91     - type: url
92       target: https://github.com/sky-org-rjoNE/catalog/blob/main/catalog-info.yaml
93     - type: url
94       target: https://github.com/sky-org-rjoNE/catalog/blob/main/all.yaml
95
```

Flask template

```
1 ---
2 apiVersion: scaffolder.backstage.io/v1beta3
3 # https://backstage.io/docs/features/software-catalog/descriptor-format#kind-template
4 kind: Template
5 metadata:
6   name: Flask
7   title: Flask
8   description: Hello flask app template
9 spec:
10   owner: devops
11   type: service
12
13   parameters:
14     - title: Fill in some steps
15       required:
16         - repositoryName
17         - organizationName
18   properties:
19     repositoryName:
20       title: Repository Name
21       type: string
22       description: Name of Repository
23     organizationName:
24       title: Organization Name
25       type: string
26       description: Name of Organization
27     enum:
28       - sky-org-rjoNE
```

```
30 steps:
31   You, now + Uncommitted changes
32
33   - id: fetch-base
34     name: Fetch Base
35     action: fetch:template
36     input:
37       url: ./content
38       values:
39         name: ${ parameters.name }
40
41   - action: catalog:write
42     id: create-catalog-info-file
43     name: Create catalog file
44     input:
45       entity:
46         apiVersion: backstage.io/v1alpha1
47         kind: Component
48         metadata:
49           name: ${ values.name | dump }
50           description: Description of flask
51           annotations:
52             github.com/project-slug: ${ parameters.organizationName + "/" + parameters.repositoryName }
53             backstage.io/techdocs-ref: dir:.
54         spec:
55           type: service
56           owner: developers
57           lifecycle: production
58
59   - id: publish
60     name: Publish
61     action: publish:github
62     input:
63       allowedHosts: ['github.com']
64       description: This is ${ parameters.name }
65       repoUrl: ${ "github.com?owner=" + parameters.organizationName + "&repo=" + parameters.repositoryName }
66
67   - id: register
68     name: Register
69     action: catalog:register
70     input:
71       repoContentsUrl: ${ steps['publish'].output.repoContentsUrl }
72       catalogInfoPath: '/catalog-info.yaml'
73
74   - id: log-message
75     name: Log Message
76     action: debug:log
77     input:
78       message: ${ "github.com?owner=" + parameters.organizationName + "&repo=" + parameters.repositoryName }
```

catalog/templates/flask/*/demo.yaml

Flask CI/CD

```
1  name: Deploy flask app
2  on:
3    push:
4      branches:
5        - master
6  jobs:
7    build:
8      runs-on: ubuntu-latest
9      steps:
10     - name: Checkout repository
11       uses: actions/checkout@v2
12
13     - name: Build docker image
14       run: cat Dockerfile
15
16     - name: gke authenticate
17       run: echo "Autheticationg to GKE"
18
19     - name: Deploy
20       run: Deploying
21
```

Demonstration

Questions?