



kubernetes



Vue CLI 3.0



# Make seamless development on cloud with kubernetes



By Praparn Luengphoonlap  
Email: [praparn@opcellent.com](mailto:praparn@opcellent.com)

Seamless development on cloud with K8S



# Agenda

- Container's trend on real-world 2018
- How development environment going today?
- Kubernetes for all...
- Make devspace seamless with "DevSpace"
- Demo Case: Vuejs3.0 (CLI) with K8S (Front-End)
- Demo Case: Python realtime with K8S (Back-End)
- Q&A

# Resource

- <https://github.com/praparn/CodeMania2018>

praparn / CodeMania2018

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

CodeMania2018 Edit

Manage topics

5 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

praparn 20181014034450	Latest commit c783a7f a day ago
Demo_Python	20181014024720 2 days ago
Demo_VueJS	20181014024720 2 days ago
Resource	20181014024720 2 days ago
README.md	20181014034450 a day ago
instruction.txt	20181014024720 2 days ago

README.md

## CodeMania2018: Make seamless development on cloud with kubernetes

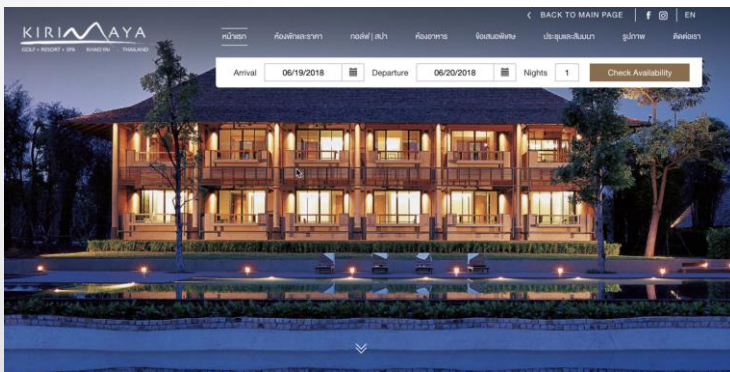
Reference tool from "devspace" :<https://github.com/covexo/devspace/blob/master/README.md>

Program inside any Kubernetes cluster (e.g. minikube, self-hosted or cloud platform) and:

iterate quickly: no more building and pushing images on every change, use hot reloading instead (e.g. with nodemon) keep your existing workflow and tools: the DevSpace CLI works with every IDE (no plugins required) access cluster-internal services and data during development debug efficiently with port forwarding and terminal proxying migrate to Docker & Kubernetes within minutes

Seamless development on cloud with K8S

# Who are we ? (Opcellent)



Seamless development on cloud with K8S



# Container's trend on real-world 2018



Seamless development on cloud with K8S

# Container's trend on real-world 2018

- Reference from “2018 Docker Usage Report” of sysdig
- Sampling from 90,000 container over company of mid-market to large enterprise
- Scope on
  - North America
  - Latin America
  - EMEA (Europe, Middle East and Africa)
  - Asia Pacific
- <https://sysdig.com/blog/2018-docker-usage-report/>

**Sysdig**

## 2018 Docker Usage Report.

An inside look at shifting container usage trends.

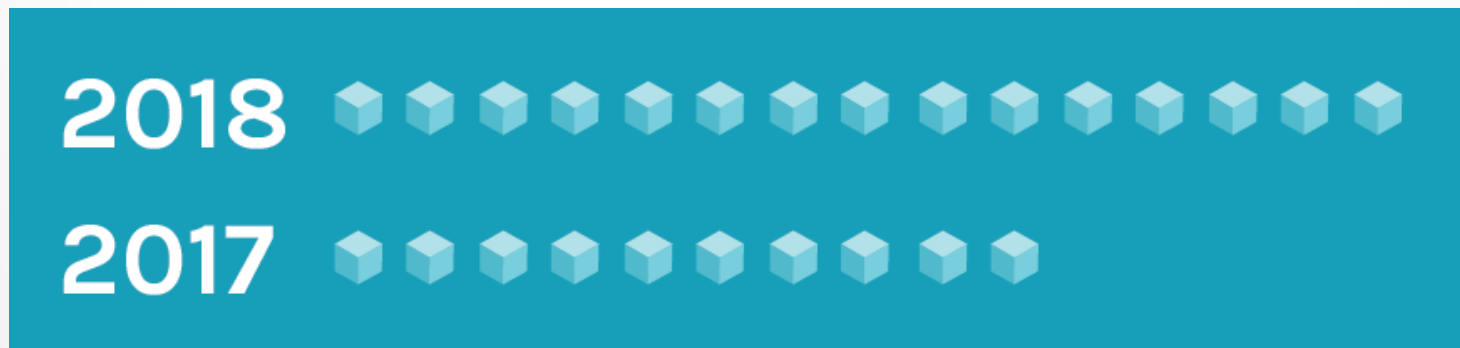
Second annual Docker Usage Report shows densities increasing, growing diversity in container runtimes.

# Container's trend on real-world 2018

- Most application component on container



- Median container density per host rises 50% (Per year)



# Container's trend on real-world 2018

- Max container housing per single host is 154 containers !!! (2017: 95) (Docker Native)

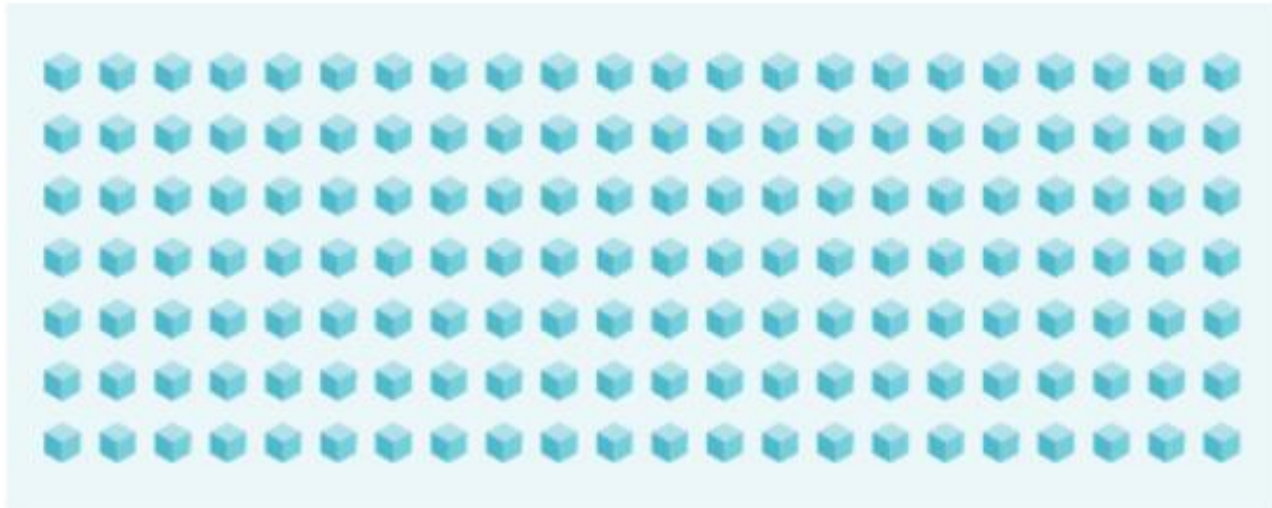
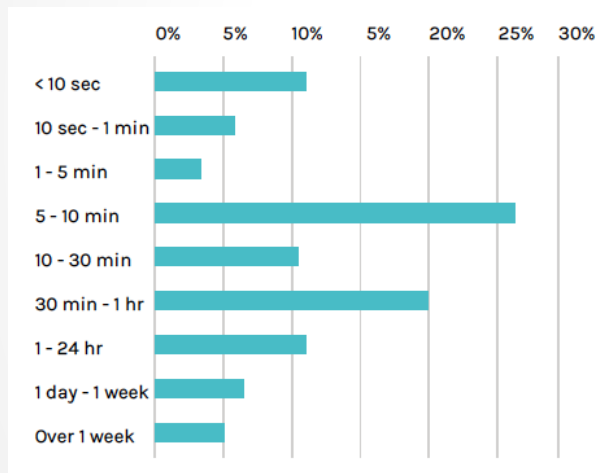


Figure 3. Max density observed: 154 containers

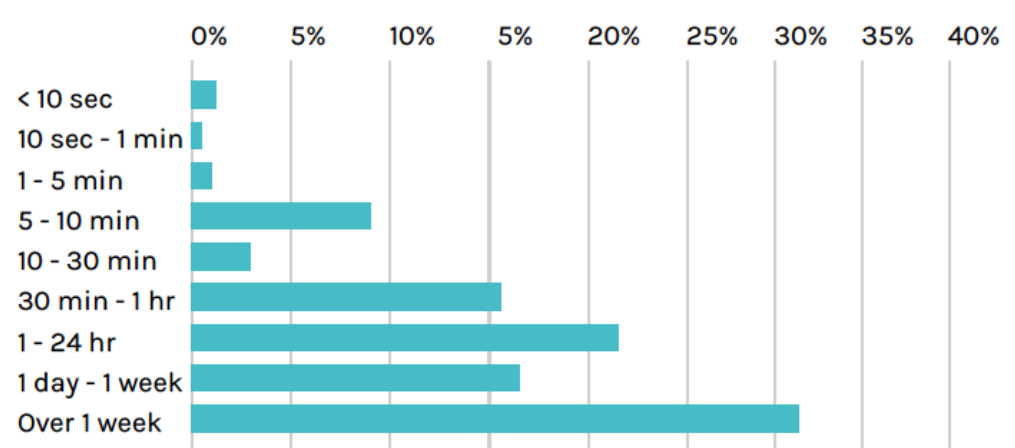


# Container's trend on real-world 2018

- Lifespan of containers and service (95% less than a week)
- (Container Run Time)



(Image Live Time)

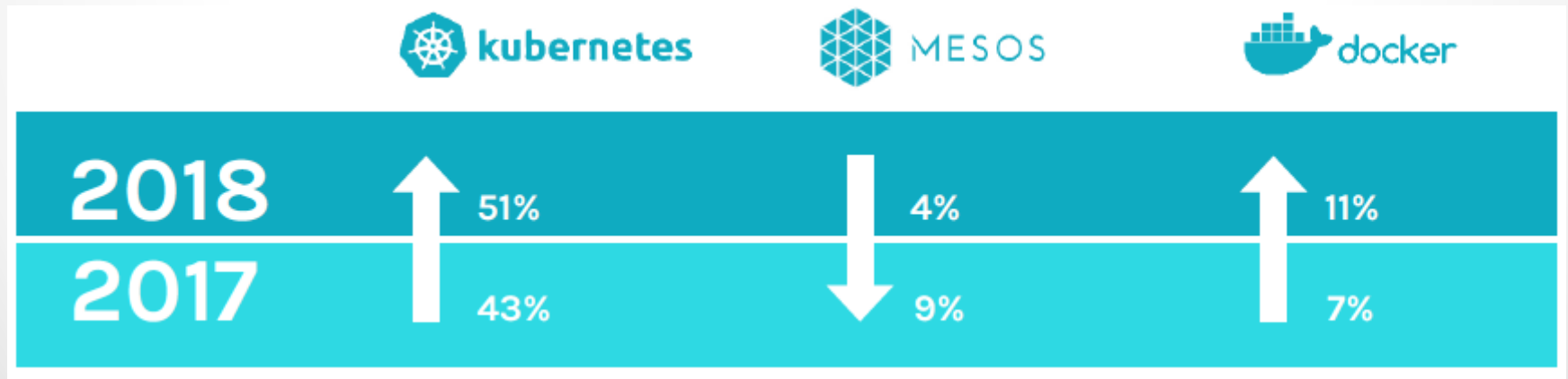


# Container's trend on real-world 2018

- Container runtime used (Almost also docker)



- Orchestrator trend (Kubernetes king of orchestrator)



# Container's trend on real-world 2018

- Conclusion
  - More company migrate their culture to “DevOps” and deploy their application from monolith to microservice with container technology (50% from 2017)
  - Application development life cycle come more faster
    - monthly → week
    - Week → less than week (70% is on this point)
  - Docker will best for container runtime and native single host
  - Kubernetes still “king of orchestrator” when running container farm

# How development environment going today



Seamless development on cloud with K8S

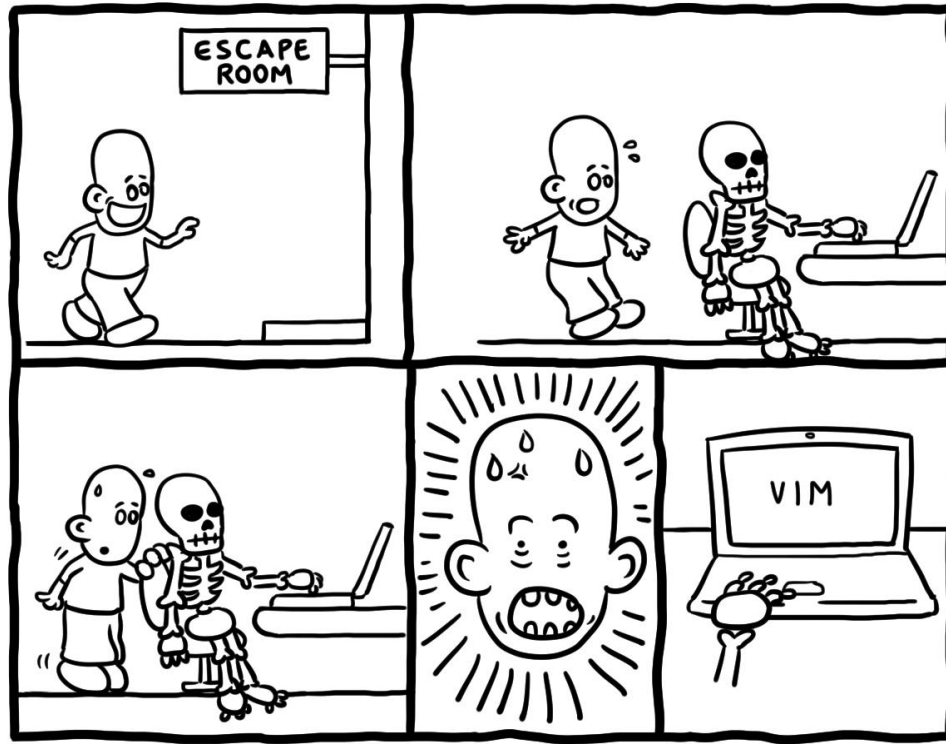


# How development env. going today

- Normally we are happy to develop on our machine with our favorite IDE (Atom, Notepad++, VSCode, NetBeans, Android Studio etc) and debug on our machine
- Many and many debug / develop until finish module
- But...
  - Some environment are available on
    - Development Server
    - Cloud Server
  - How to synchronize code/configure ?
  - No IDE tool...Oh ! We have Vim
  - ...



# How development env. going today



Daniel Stori {turnoff.us}



**I Am Devloper**

@iamdevloper

Following

I've been using Vim for about 2 years now, mostly because I can't figure out how to exit it.

RETWEETS  
**14,083**

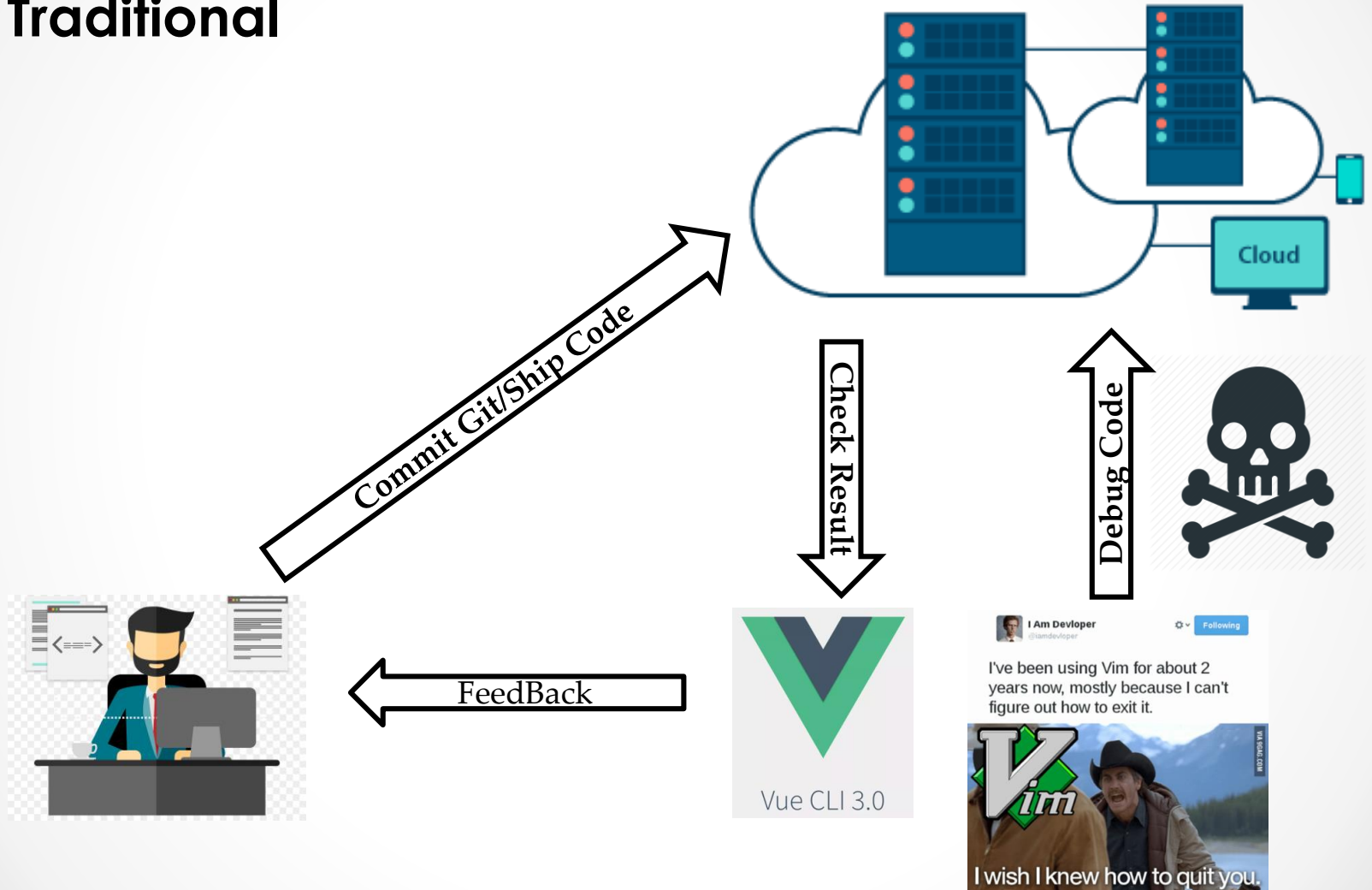
LIKES  
**8,154**



Seamless development on cloud with K8S

# How development env. going today

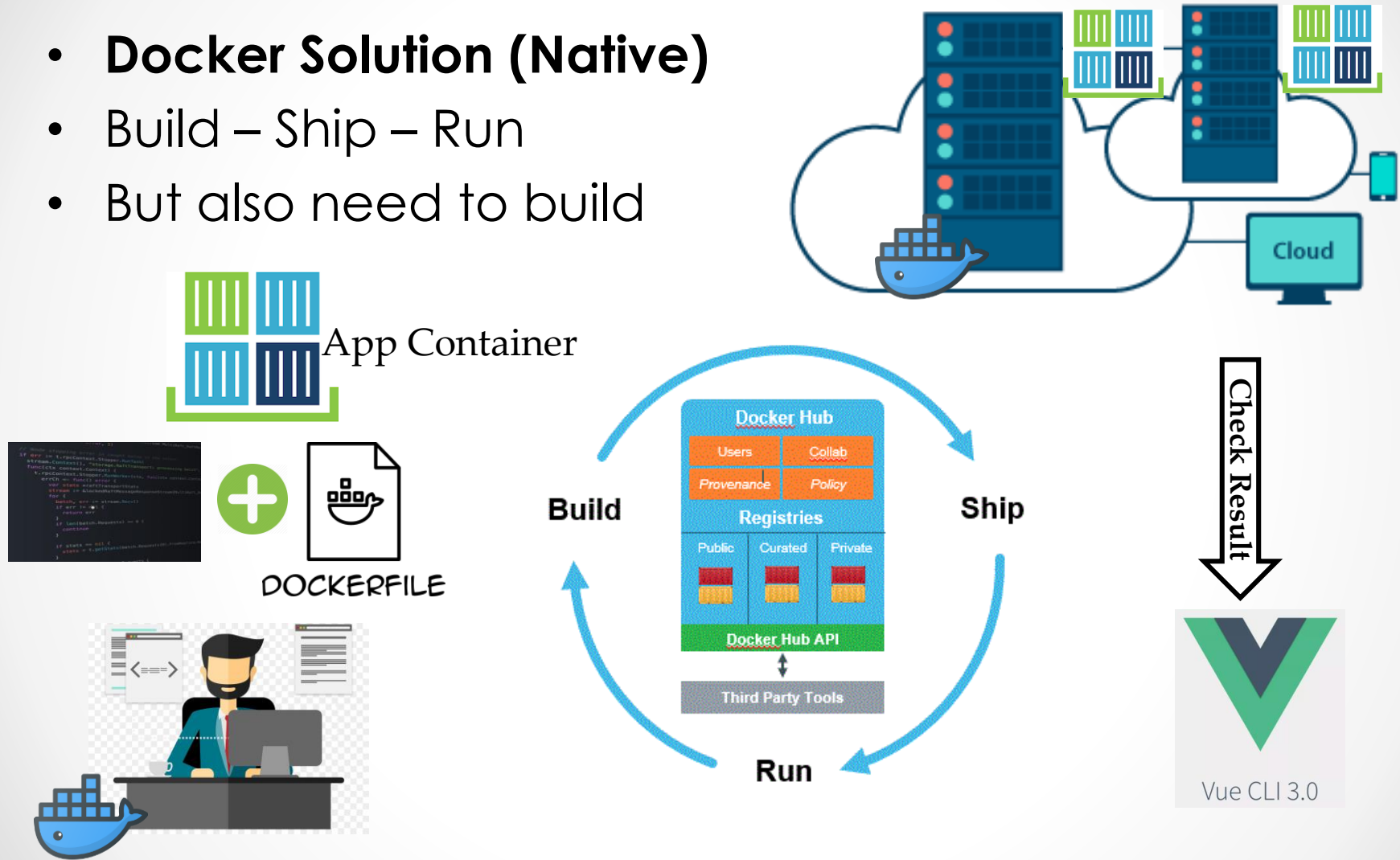
- Traditional



Seamless development on cloud with K8S

# How development env. going today

- **Docker Solution (Native)**
- Build – Ship – Run
- But also need to build



Seamless development on cloud with K8S



# Kubernetes for All



Seamless development on cloud with K8S

# Kubernetes for All

- Docker native is good for container runtime...
- But...Kubernetes is better orchestrator for cluster system
- Kubernetes will short name with “K8S”
- Docker is easier for developer to deploy application just “docker container run ....” all of it will appear like magic  
^^
- But this not easy like that on “K8S” T\_T
  - Kubernetes will operate via “YAML” for almost of it. So we need to create YAML file also (And change it everytime we had been change code !!!)
  - Kubernetes will separate workload and service from each other. So YAML file also need to separate ???

# Kubernetes for All

- Docker native is good for container runtime...
- But...Kubernetes is better orchestrator for cluster system
- Kubernetes will short name with “K8S”
- Docker is easier for developer to deploy application just “docker container run ....” all of it will appear like magic  
^^
- But this not easy like that on “K8S” T\_T
  - Kubernetes will operate via “YAML” for almost of it. So we need to create YAML file also (And change it everytime we had been change code !!!)
  - Kubernetes will separate workload and service from each other. So YAML file also need to separate ???

# Kubernetes for All

Topic	K8S	Docker/Swarm
Architecture	Open-system (Base on cluster manager "Borg" for support complex workload)	<b>Swarm:</b> Proprietary of Docker product, "Easy to use", "Extend capability of Docker in cluster"
Operation command	Almost operate by "YAML" file (Declarative Command)	Almost operate by "command" (Imperative Command)
Unit of Work	Pods (Pods >= Container)	Container
How to Identify Work	"Label operation"	<b>Docker:</b> By container name <b>Swarm:</b> By service/stack name
Level of workload management	Service Level: (Simple) Replication Level: (Auto healing) Deployment Level: (Auto healing + Roll Update)	<b>Docker:</b> N/A <b>Swarm:</b> Service Level (Snag with service/stack)
Auto scaling	HPA (Horizontal Pods Scaling) base on CPU	No
Health check	Liveness & Readiness (Multi option to check application health)	Service health only

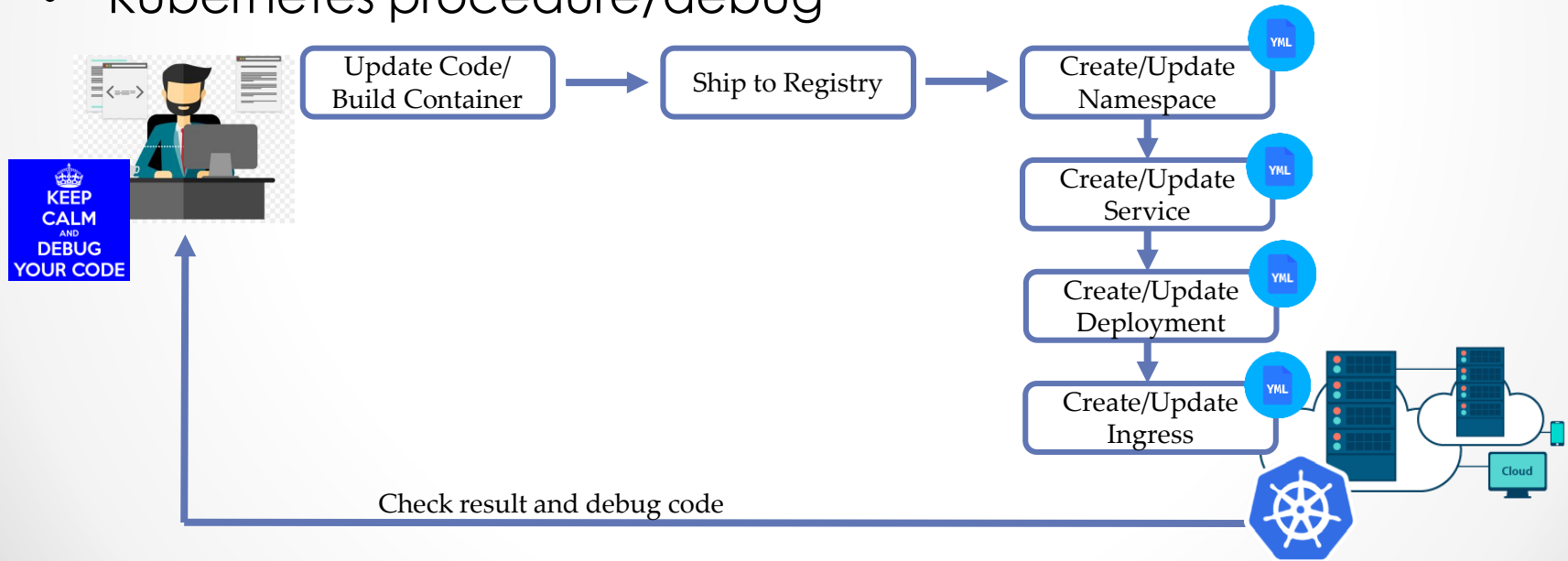


# Kubernetes for All

- Docker procedure/debug



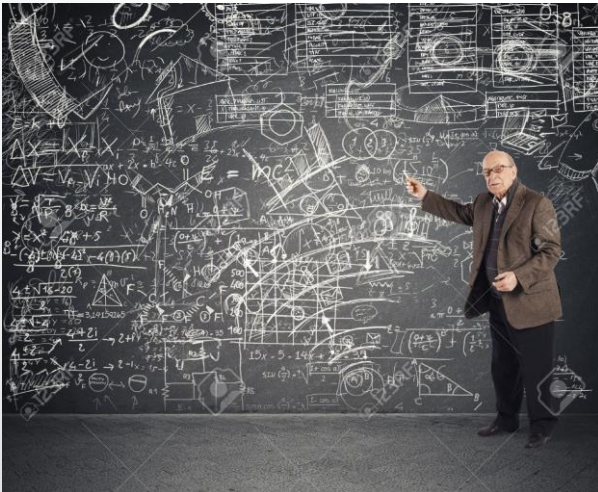
- Kubernetes procedure/debug



Seamless development on cloud with K8S

# Kubernetes for All

- But...We just want to debug our code and make surly it can survive without any error ^^
- We just need
  - Some simple ...
  - Some sync code often as need...
  - Some automatic operate easy
  - ...
- We have a lot of problem to solve TT...



Seamless development on cloud with K8S

# Make devspace seamless with “DevSpace”



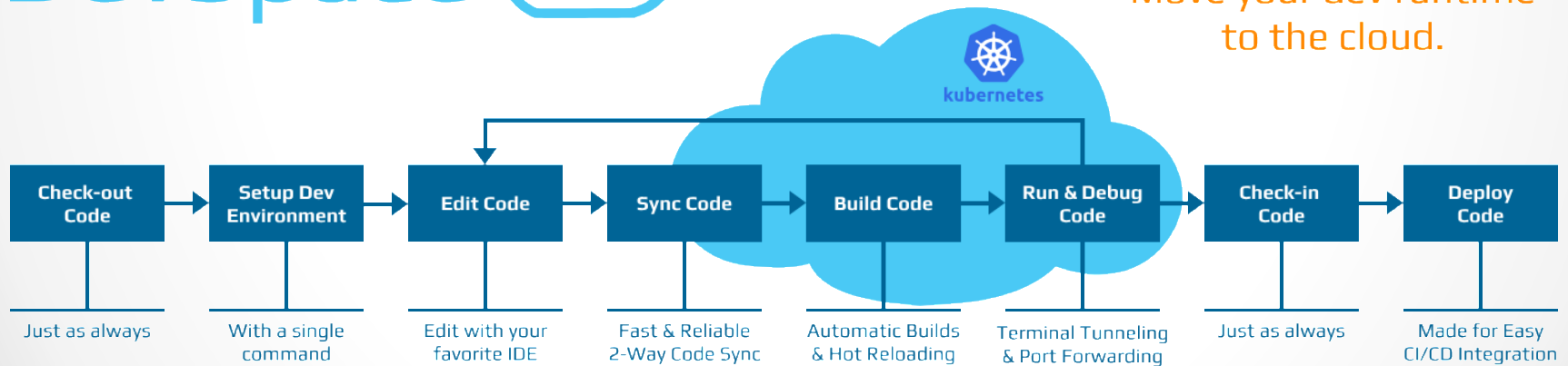
Seamless development on cloud with K8S

# Make seamless dev with devspace

- Let's make development seamless with "DevSpace"
  - DevSpace will provide tools for make development better in Kubernetes
  - Code was develop on your machine but sync with K8S !!!

## DevSpace

Move your dev runtime to the cloud.



Ref: <https://github.com/covexo/devspace/blob/master/README.md>

Seamless development on cloud with K8S

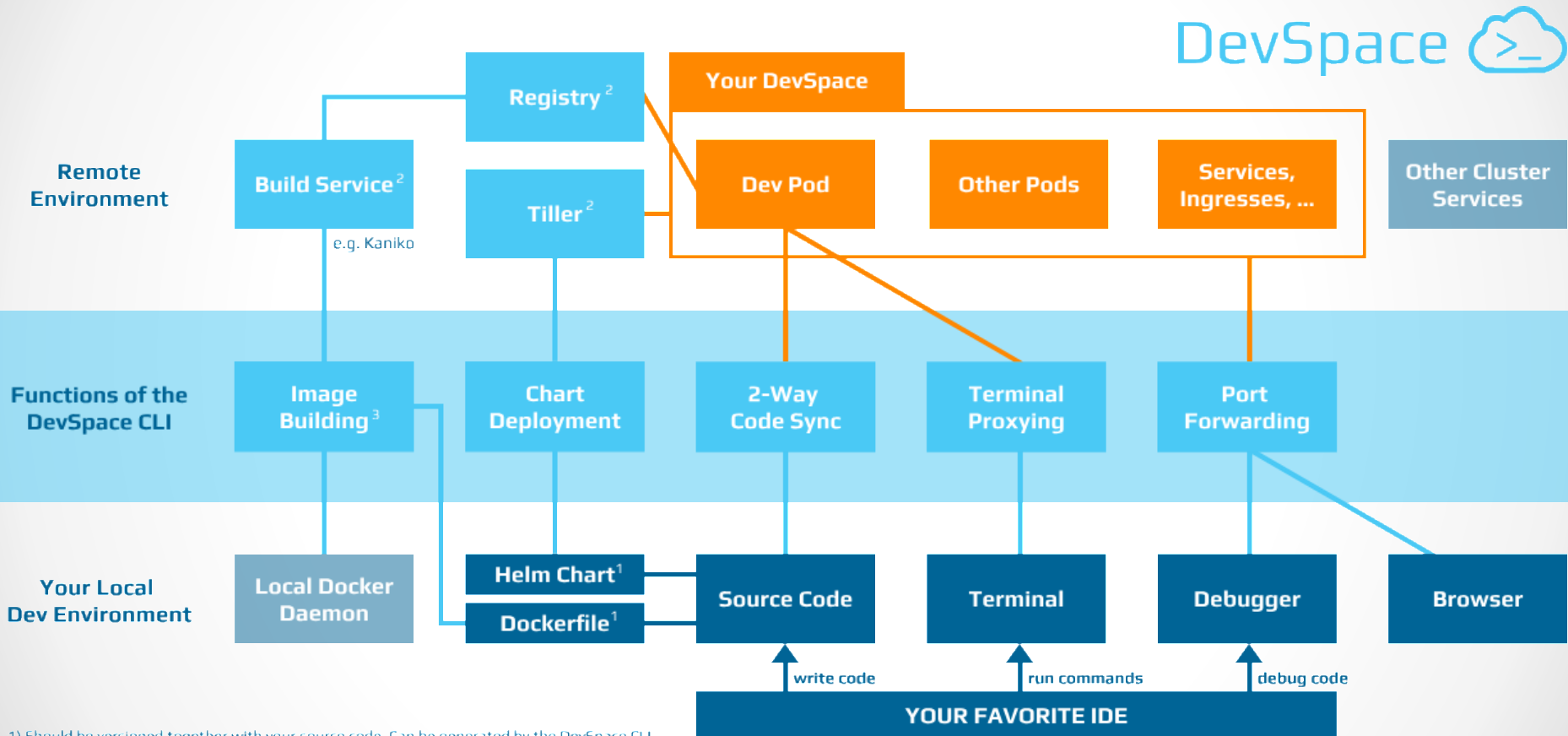


# Make seamless dev with devspace

- Feature
  - **Iterate quickly:** no more building and pushing images on every change, use hot reloading instead
  - **Keep your existing workflow and tools:** the DevSpace CLI works with every IDE
  - **Access cluster-internal:** access services and data during development debug efficiently with port forwarding and terminal proxying
  - **Migrate:** Docker → Kubernetes within minutes
- Support Platform
  - Mac OS X
  - Linux
  - Windows

# Make seamless dev with devspace

- Architecture



1) Should be versioned together with your source code. Can be generated by the DevSpace CLI.

2) The DevSpace CLI can automatically deploy and manage these services within your cluster.

3) You can either use your local docker daemon to build the Dockerfile or use the in-cluster build service provided through Kaniko.

# Make seamless dev with devspace

- Feature
  - **Iterate quickly:** no more building and pushing images on every change, use hot reloading instead
  - **Keep your existing workflow and tools:** the DevSpace CLI works with every IDE
  - **Access cluster-internal:** access services and data during development debug efficiently with port forwarding and terminal proxying
  - **Migrate:** Docker → Kubernetes within minutes
- Support Platform for developer's machine
  - Mac OS X
  - Linux
  - Windows

# Make seamless dev with devspace

- Support Platform for kubernetes's farm
  - Minikube
  - Docker for Mac/Docker for Windows (Enable K8S)
  - Kubernetes on Server (Cloud/On-Prem)
- Single command for create all environment
  - devspace up/down: build environment/start-stop
  - devspace and connect to environment
  - devspace enter: jump to environment as need
  - devspace reset: clear entire project environment
  - devspace add/remove (package/sync/port)
  - devspace status
  - etc

# Make seamless dev with devspace

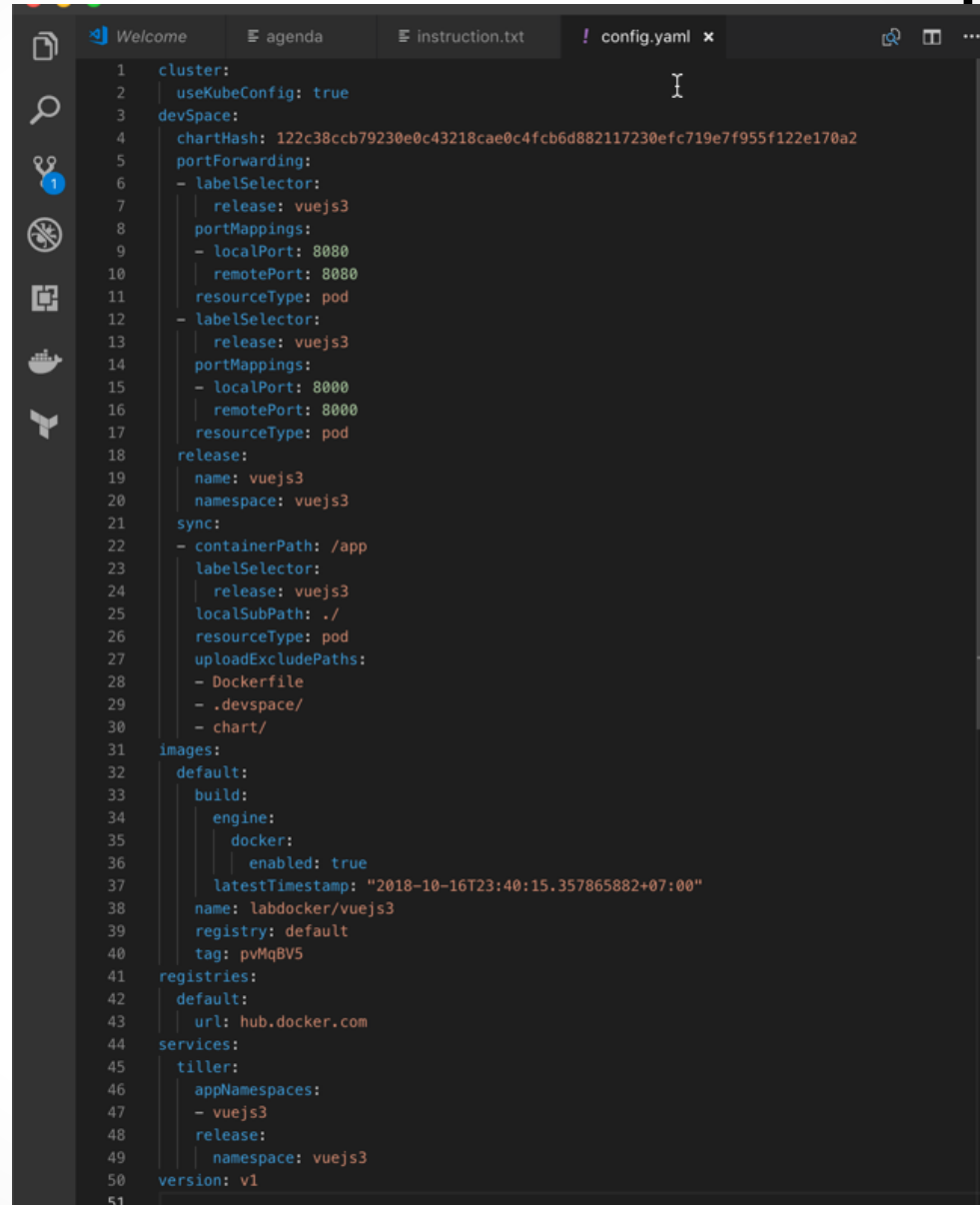
- Folder structure

```
YOUR_PROJECT_PATH/  
|  
|-- Dockerfile  
|  
|-- chart/  
|   |-- Chart.yaml  
|   |-- values.yaml  
|   |-- templates/  
|       |-- deployment.yaml  
|       |-- service.yaml  
|       |-- ingress.yaml  
|  
|-- .devspace/  
|   |-- .gitignore  
|   |-- cluster.yaml  
|   |-- config.yaml
```



# Make seamless dev with devspace

- config.yaml



```
1 cluster:
2   useKubeConfig: true
3 devspace:
4   chartHash: 122c38ccb79230e0c43218cae0c4fcb6d882117230efc719e7f955f122e170a2
5   portForwarding:
6     - labelSelector:
7         release: vuejs3
8       portMappings:
9         - localPort: 8080
10           remotePort: 8080
11       resourceType: pod
12     - labelSelector:
13         release: vuejs3
14       portMappings:
15         - localPort: 8000
16           remotePort: 8000
17       resourceType: pod
18   release:
19     name: vuejs3
20     namespace: vuejs3
21   sync:
22     - containerPath: /app
23       labelSelector:
24         release: vuejs3
25       localSubPath: ./
26       resourceType: pod
27       uploadExcludePaths:
28         - Dockerfile
29         - .devspace/
30         - chart/
31   images:
32     default:
33       build:
34         engine:
35           docker:
36             enabled: true
37           latestTimestamp: "2018-10-16T23:40:15.357865882+07:00"
38           name: labdocker/vuejs3
39           registry: default
40           tag: pvMqBV5
41   registries:
42     default:
43       url: hub.docker.com
44   services:
45     tiller:
46       appNamespaces:
47         - vuejs3
48       release:
49         namespace: vuejs3
50 version: v1
51
```

Seamless development on cloud with K8S

# Make seamless dev with devspace

- Example

```
praparns-MacBook-Pro:Demo_VueJS praparn$ devspace up
What is the major programming language of your project?
Supported languages: csharp, go, java, javascript, none, php, python, ruby, typescript
Press ENTER to use: none
> javascript
Do you want to use your existing $HOME/.kube/config for Kubernetes access? (yes | no)
Press ENTER to use: yes
>
What is the name of your application?
Press ENTER to use: devspace
> vuejs3
Which port(s) does your application listen on? (separated by spaces)
> 8080 8000
Which Kubernetes namespace should your application run in?
Press ENTER to use: default
> vuejs3
Which Kubernetes namespace should your tiller server run in?
Press ENTER to use: vuejs3
>
Should we create a private registry within your Kubernetes cluster for you? (yes | no)
Press ENTER to use: no
>
Which registry do you want to push to? ('hub.docker.com' or URL)
Press ENTER to use: hub.docker.com
>
Which image name do you want to use on Docker Hub?
Press ENTER to use: labdocker/vuejs3
>

[WARN] Unable to check permissions: If you run into errors, please create the ClusterRoleBinding 'devspace-users' as described here: https://devspace.covexo.com/docs/advanced/rbac.html
[DONE] ✓ Tiller started
[DONE] ✓ Initialized helm client
[INFO] Building image 'default' with engine 'docker'
[DONE] ✓ Authentication successful (hub.docker.com)
```

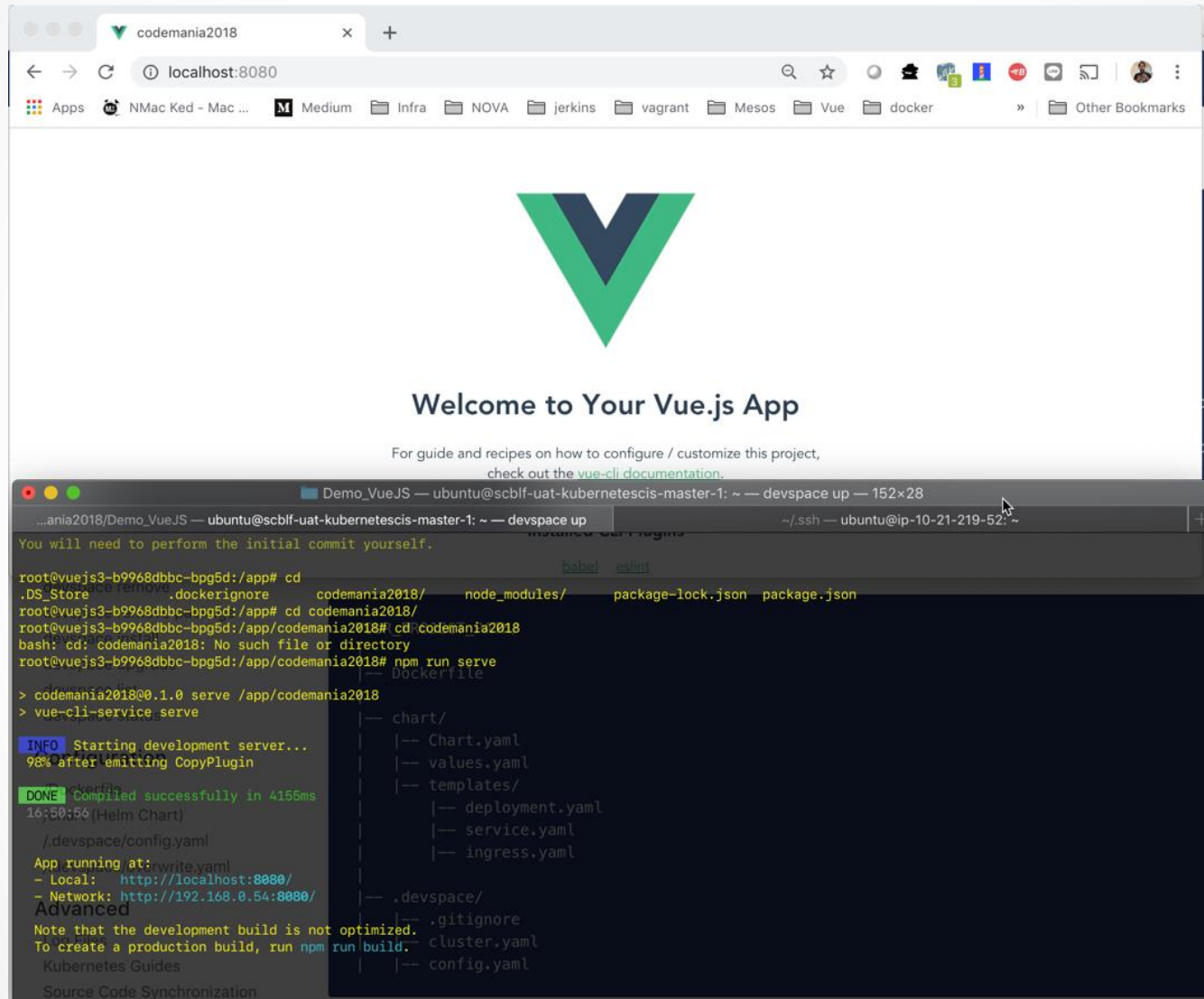
Seamless development on cloud with K8S

# Demo Case:



Seamless development on cloud with K8S

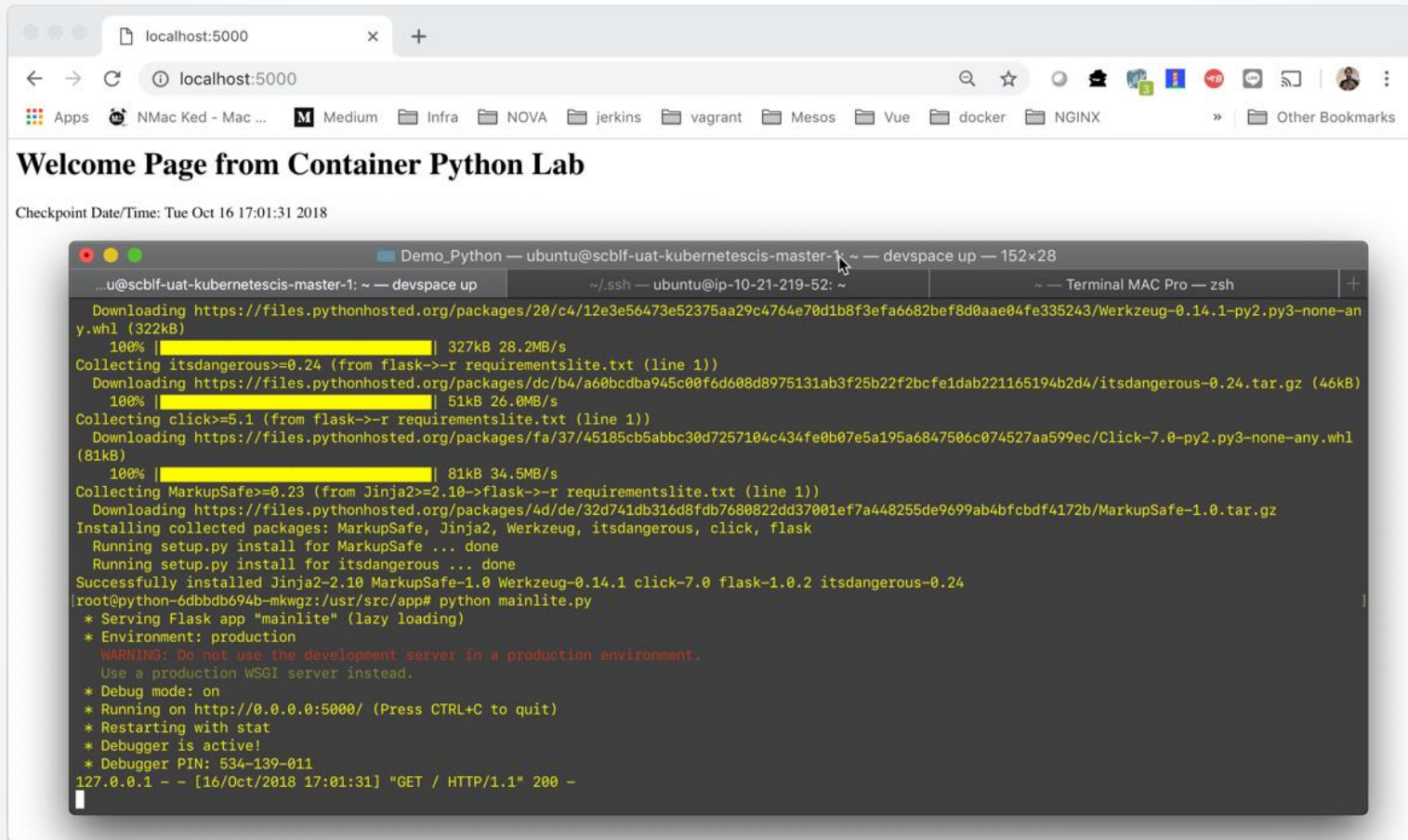
# Demo Case: Vuejs3.0 (CLI)



Seamless development on cloud with K8S



# Demo Case: Python with K8S



The screenshot shows a web browser window with the address bar set to `localhost:5000`. The page title is "Welcome Page from Container Python Lab" and the content includes a timestamp: "Checkpoint Date/Time: Tue Oct 16 17:01:31 2018". Below the page content is a terminal window titled "Demo\_Python — ubuntu@scblf-uat-kubernetescis-master-1 ~ — devspace up — 152x28". The terminal output shows the installation of a Flask application with the following steps:

```
Downloading https://files.pythonhosted.org/packages/28/c4/12e3e56473e52375aa29c4764e70d1b8f3efa6682bef8d0aae04fe335243/Werkzeug-0.14.1-py2.py3-none-any.whl (322kB)
100% |#####| 327kB 28.2MB/s
Collecting itsdangerous>=0.24 (from flask->r requirementslite.txt (line 1))
Downloading https://files.pythonhosted.org/packages/dc/b4/a60bcd9a945c00f6d608d8975131ab3f25b22f2bcfe1dab221165194b2d4/itsdangerous-0.24.tar.gz (46kB)
100% |#####| 51kB 26.0MB/s
Collecting click>=5.1 (from flask->r requirementslite.txt (line 1))
Downloading https://files.pythonhosted.org/packages/fa/37/45185cb5abbc30d7257104c434fe0b07e5a195a6847506c074527aa599ec/Click-7.0-py2.py3-none-any.whl (81kB)
100% |#####| 81kB 34.5MB/s
Collecting MarkupSafe>=0.23 (from Jinja2>=2.10->flask->r requirementslite.txt (line 1))
Downloading https://files.pythonhosted.org/packages/4d/de/32d741db316d8fdb7680822dd37001ef7a448255de9699ab4bfcdbf4172b/MarkupSafe-1.0.tar.gz
Installing collected packages: MarkupSafe, Jinja2, Werkzeug, itsdangerous, click, flask
Running setup.py install for MarkupSafe ... done
Running setup.py install for itsdangerous ... done
Successfully installed Jinja2-2.10 MarkupSafe-1.0 Werkzeug-0.14.1 click-7.0 flask-1.0.2 itsdangerous-0.24
root@python-6dbdb694b-mkwgz:/usr/src/app# python mainlite.py
* Serving Flask app "mainlite" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 534-139-011
127.0.0.1 - - [16/Oct/2018 17:01:31] "GET / HTTP/1.1" 200 -
```

Seamless development on cloud with K8S





# kubernetes



Vue CLI 3.0



## Q&A

By Praparn Luengphoonlap  
Email: [praparn@opcellent.com](mailto:praparn@opcellent.com)

Seamless development on cloud with K8S

