Crane

An easy and elegant way to build and manage your notebook images

Kaun Lin Huang (@kentwelcome), InfuseAI
Who Am I?
Kaun Lin Huang
a.k.a. Kent Huang
<kent@infuseai.io>
SWE @ InfuseAI
🌐 @kentwelcome
Why
Why we build Crane?
The problem we face

- Building a container image is easy for a developer, but not easy for others
- Writing Dockerfiles from scratch is annoying
  - Should update repo before installing applications
  - Should cleanup cache after installing applications
  - All Update, Install, Cleanup actions should be run in a same command
- Need to manually change the tag when pushing to remote registry
Why we built Crane?
Problem from our Users

• Who are the Users?
  • Data scientist, ML researcher
• Users who aren’t familiar with using command-line interface
• Users who don’t have strong background knowledge of container technology
Real-World Use Cases
Setup All Workshop Dependencies

*Note: This Notebook Will Take A Few Minutes To Complete.*

**Please Be Patient.**

```
In [1]:
    python --version
```

```
In [2]:
    pip list
```

**Pip**

```
In [3]:
    pip install --disable-pip-version-check -q pip --upgrade > /dev/null
    pip install --disable-pip-version-check -q wrapt --upgrade > /dev/null
```

**AWS CLI and AWS Python SDK (boto3)**

```
In [4]:
    pip install --disable-pip-version-check -q awscli==1.21.264 boto3==1.16.56 botocore==1.19.56
```

**SageMaker**

```
In [5]:
    pip install --disable-pip-version-check -q sagemaker==2.29.0
    pip install --disable-pip-version-check -q sagemaker==1.0.1
    pip install --disable-pip-version-check -q sagemaker-experiments==0.1.26
```

**PyTorch**

```
In [6]:
    conda install -y pytorch==1.6.0 -c pytorch
```

**TensorFlow**

```
In [7]:
    pip install --disable-pip-version-check -q tensorflow==2.3.1
```
Better clean up the apt cache by removing /var/lib/apt/lists to reduces the image size

Better to add --no-cache-dir when pip install
What
What’s Crane

• A minimalist container image builder
• Desktop application for Windows and macOS
• Extend an existing container image with custom apt/conda/pip packages without writing any Dockerfile
• Manage and push local container images to remote registries (ex. Dockerhub, AWS ECR)
• Inspect a container image with Image Layer information
How Crane Work

- As a user-interface of the local Docker Engine
  - Generates Dockerfile and builds the container image
  - Lists the container images
  - Inspects the selected container images by the Image Layer
  - Pushes images to remote registries
    - Dockerhub => Docker HUB API
    - AWS ECR => AWS SDK
Demo
Inside the Crane

Technical Detail

- Desktop application created by Electron
  - FrontEnd: React
  - BackEnd: Node.js
How to let Node.js access the Docker Engine?
Node.js + Docker

- Dockerode
- https://www.npmjs.com/package/dockerode
- Via Docker Engine API
Future Roadmap

- More ways to customize the container image
- Ability to download the generated Dockerfile
- Support more remote registries
  - Private registry (ex. Docker registry, Harbor, etc...)
  - GCR Registry, Azure Container Registry
Summary

- Crane is minimalist container image builder.
- Our goal is to enable more users to use containers.
  - Focus on ML researchers & Data scientists.
- Welcome to give feedback in our GitHub repo:
  - [https://github.com/InfuseAI/crane](https://github.com/InfuseAI/crane)
- Join our Discord:
  - [https://discord.gg/ZE8pQ8gRWy](https://discord.gg/ZE8pQ8gRWy)
Reference

- https://docs.docker.com/develop/develop-images/dockerfile_best-practices
- https://github.com/cyourth-cognonic/aws-workshop/blob/9758252e53b527d546f430b24171d5e4b0a051bf/oreilly_book/01_intro/01_Setup_Dependencies.ipynb
- https://github.com/CONABIO/kube_sipecam/blob/102777676f3d72380538e9e8257e2ffdd5c0fbc6/dockerfiles/audio/tf_kale/0.4.0/2.1.0/Dockerfile
- https://docs.docker.com/engine/api/sdk/examples/
Thanks

Q&A