GDG Ahmedabad WTM Ahmedabad

# Rishit Dagli

10-grade student,
past TED-X and
Ted-Ed speaker

## Deploying models to production with TensorFlow model server

**Event link:** https://www.meetup.com/GDG-Ahmedabad/events/270477738/

GDGAhmedabad GDGAhmedabad GDG-Ahmedabad

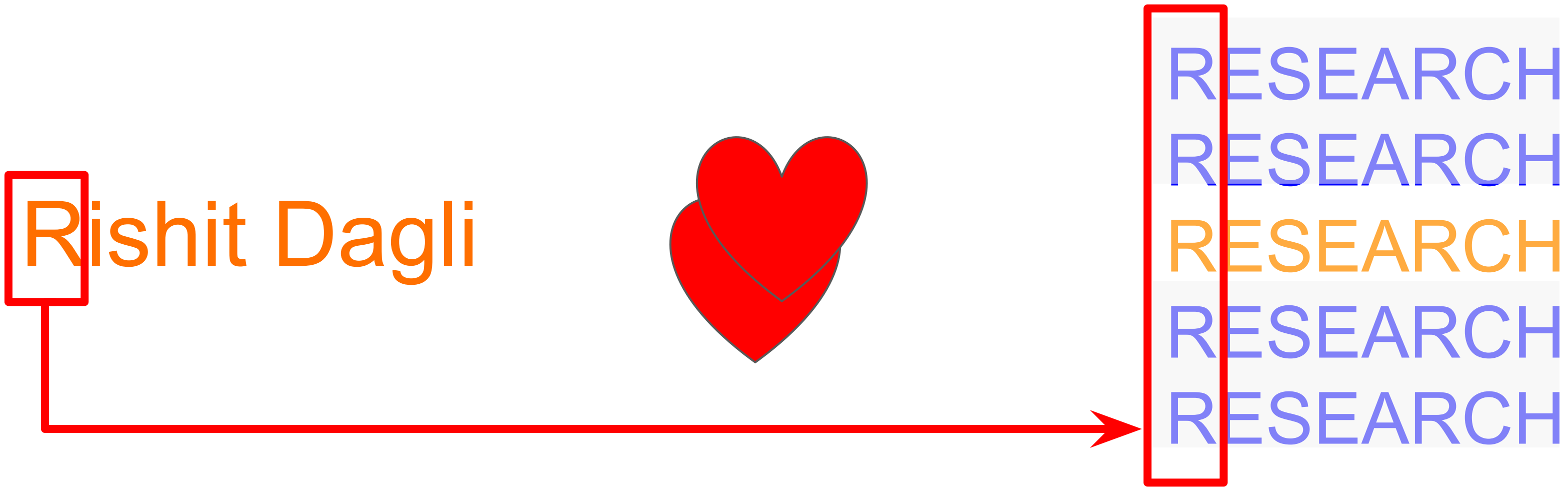
Rishit Dagli ❤ RESEARCH

RESEARCH

RESEARCH

RESEARCH

RESEARCH



Rishit Dagli

RESEARCH

RESEARCH

RESEARCH

RESEARCH

RESEARCH

GDG Ahmedabad WTM Ahmedabad

# Rishit Dagli

10-grade student,
past TED-X and
Ted-Ed speaker

## Deploying models to production with TensorFlow model server

**Event link:** https://www.meetup.com/GDG-Ahmedabad/events/270477738/

GDGAhmedabad GDGAhmedabad GDG-Ahmedabad

# Ideal Audience

- Devs who have worked on Deep Learning Models (Keras)
- Devs looking for ways to put their model into production ready manner

Motivation behind a process for deployment 01

What things to take care of? 02

What is TF Model server? 03

04 What can it do?

- Versioning
- Iaas
- CI/ CD

05 Auto Scaling

06 QnA

# Motivation behind a process for deployment

Source: memegenerator.net

Me: The UI is super simple, users will love it!

User:

Life is just ...

# What things to take care of?

# What things to take care of?

- Package the model

# What things to take care of?

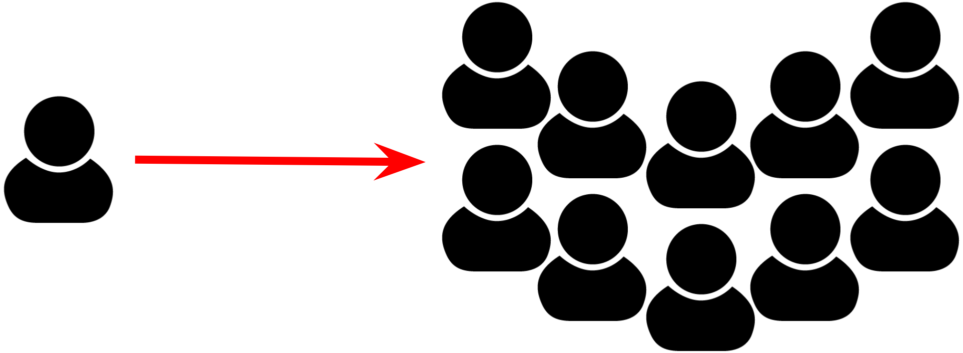
- Package the model
- Post the model on Cloud Hosted Server

# What things to take care of?

- Package the model
- Post the model on Cloud Hosted Server
- Maintain the server

# What things to take care of?

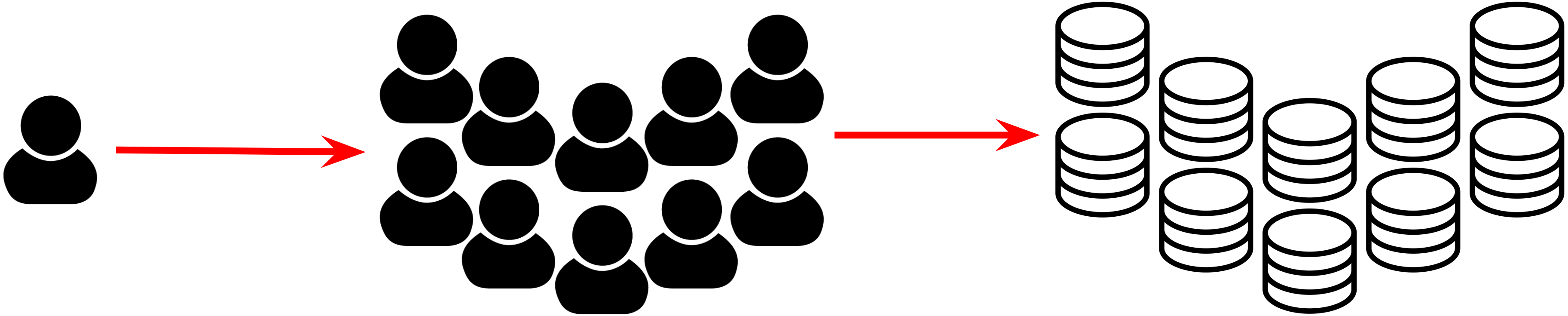
- Package the model
- Post the model on Cloud Hosted Server
- Maintain the server
  - Auto-scale



GDG Ahmedabad

# What things to take care of?

- Package the model
- Post the model on Cloud Hosted Server
- Maintain the server
    - Auto-scale



# What things to take care of?

- Package the model
- Post the model on Cloud Hosted Server
- Maintain the server
    - Auto-scale
    - Global availability

# What things to take care of?

- Package the model
- Post the model on Cloud Hosted Server
- Maintain the server
    - Auto-scale
    - Global availability
    - And many more …

# What things to take care of?

- Package the model
- Post the model on Cloud Hosted Server
- Maintain the server
  - Auto-scale
  - Global availability
  - And many more ...
- API

# What things to take care of?

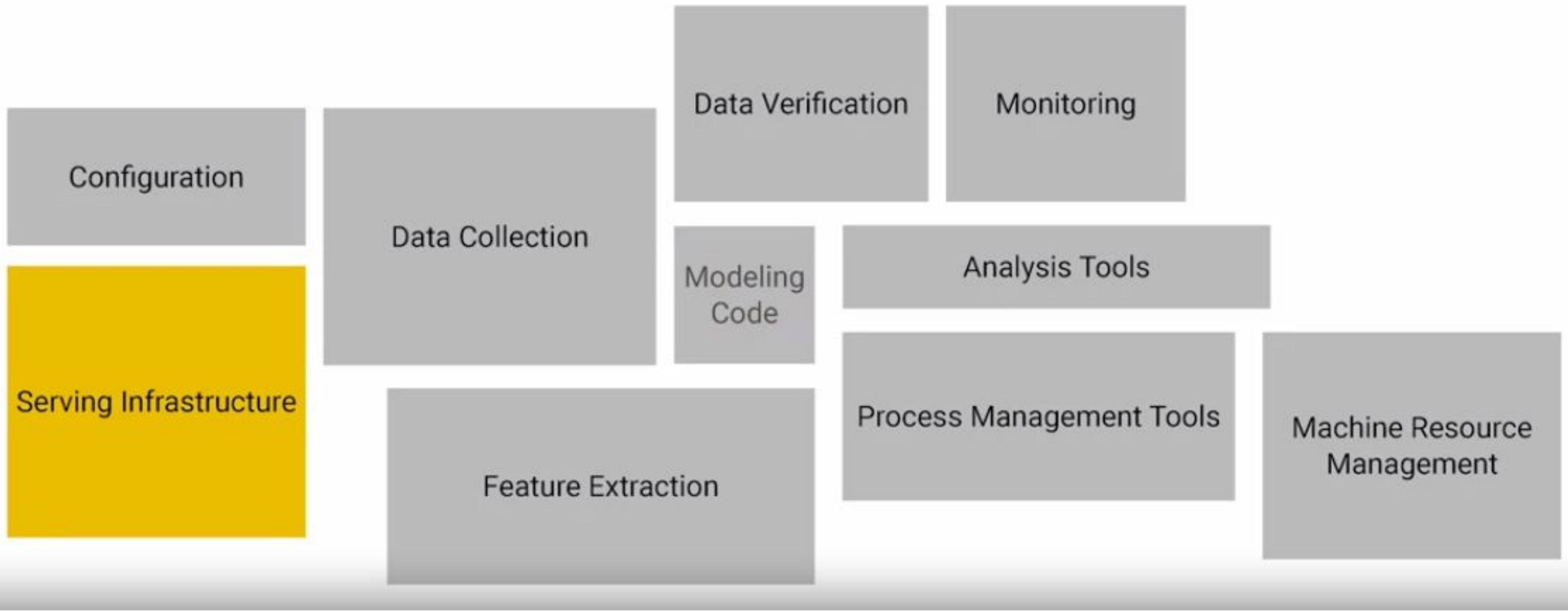
- Package the model
- Post the model on Cloud Hosted Server
- Maintain the server
  - Auto-scale
  - Global availability
  - And many more ...
- API
- Model Versioning

# What is TF Model Server?

Serving

# TensorFlow Extended

Serving



Configuration

Serving Infrastructure

Data Collection

Feature Extraction

Data Verification

Modeling Code

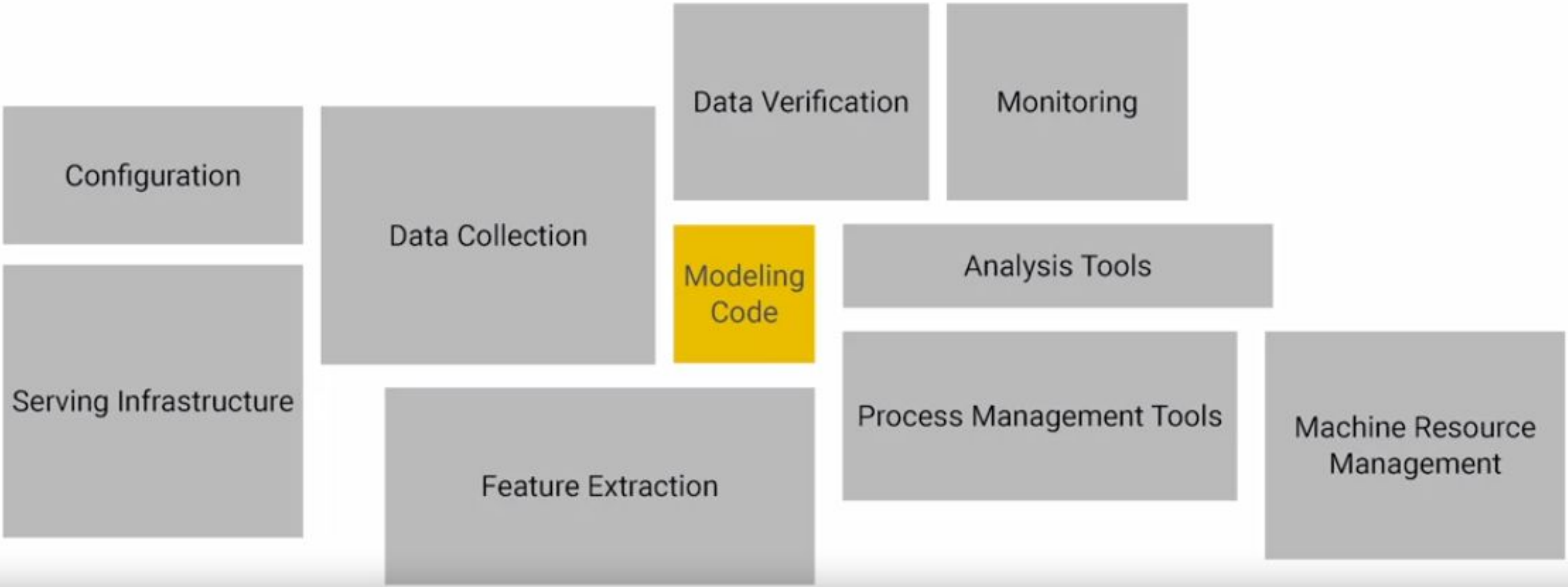
Monitoring

Analysis Tools

Process Management Tools

Machine Resource Management

Credits: @lmoroney



Data Verification

Monitoring

Configuration

Data Collection

Modeling Code

Analysis Tools

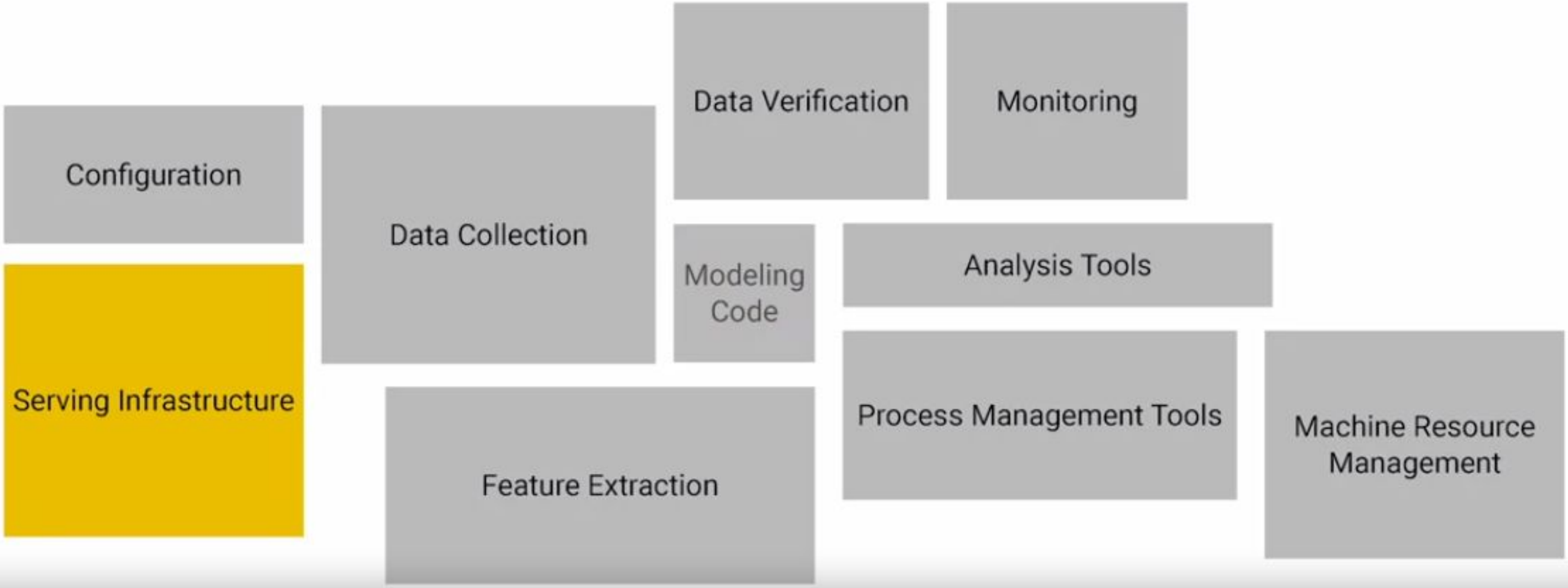
Serving Infrastructure

Process Management Tools

Machine Resource Management

Feature Extraction

Credits: @lmoroney



Data Verification

Monitoring

Configuration

Data Collection

Modeling Code

Analysis Tools

Serving Infrastructure

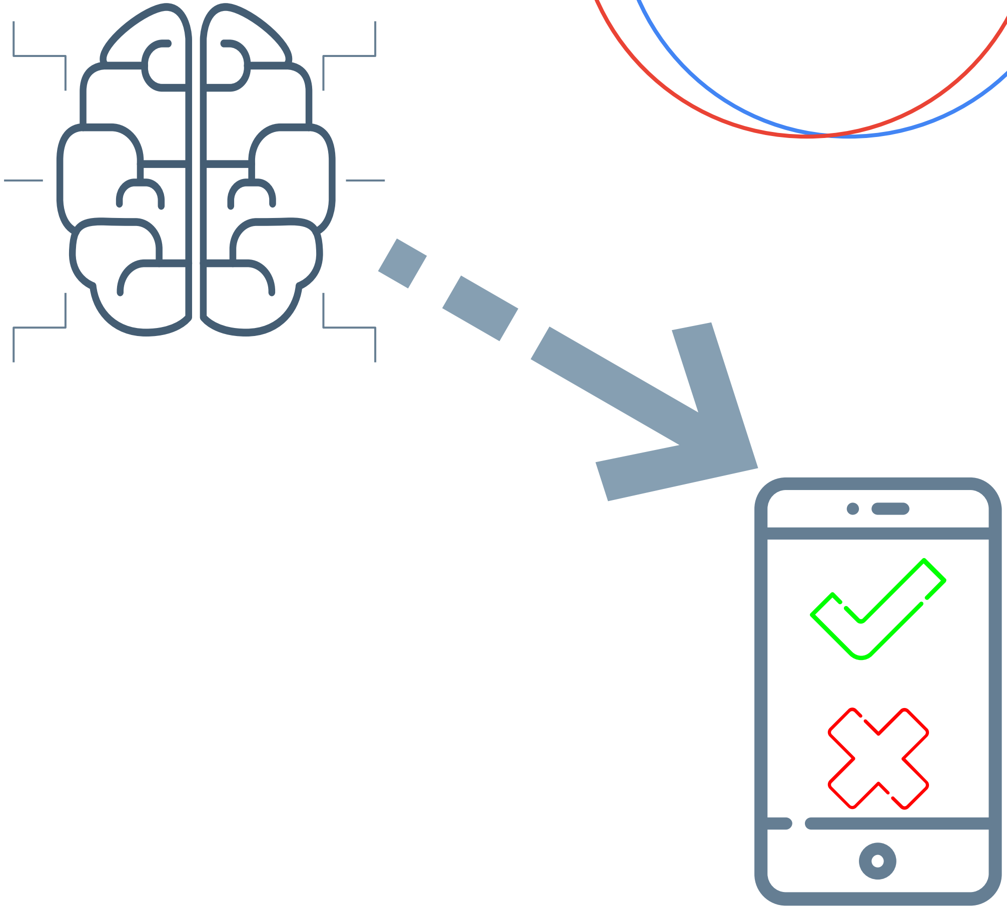
Process Management Tools

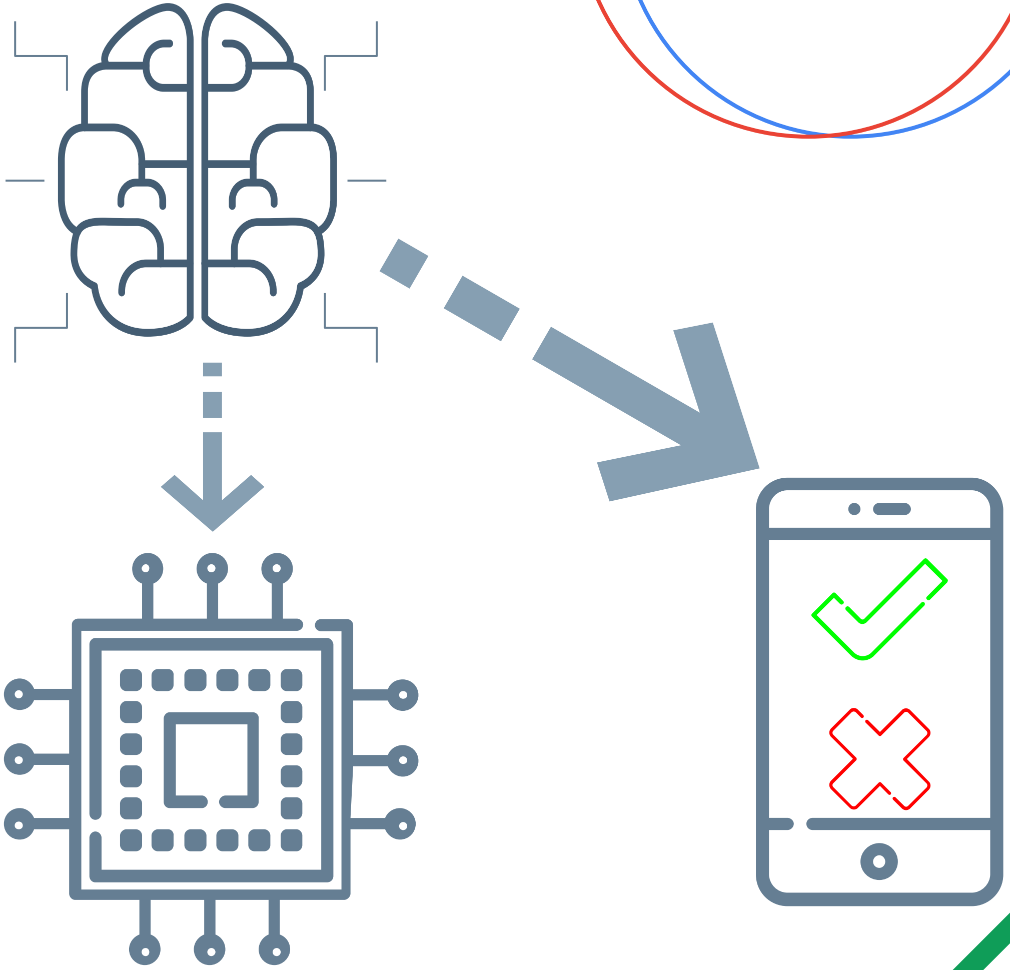
Machine Resource Management

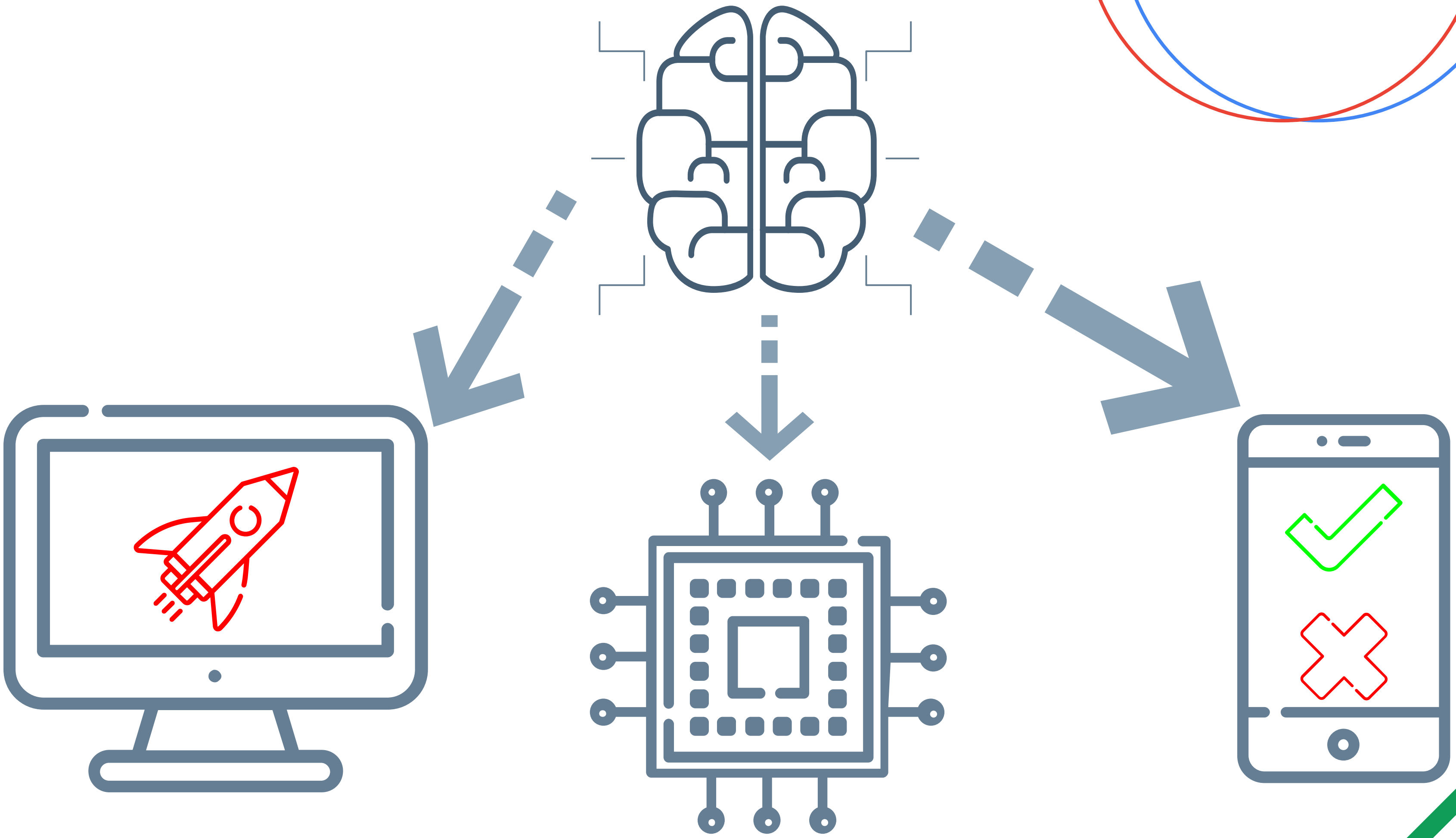
Feature Extraction

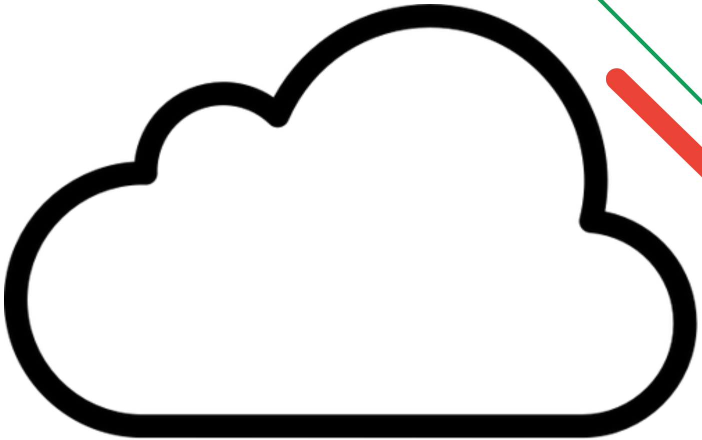
Credits: @lmoroney

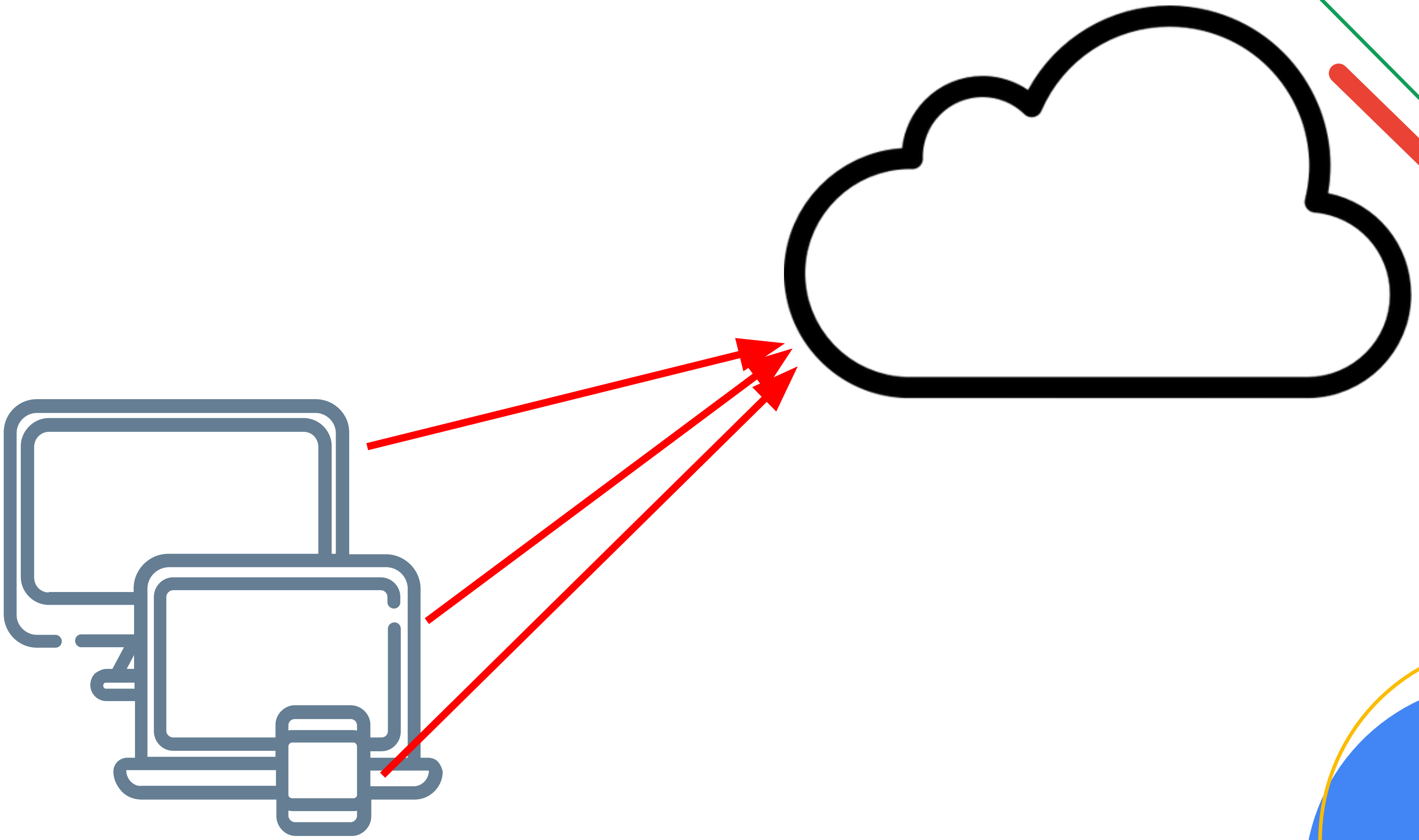
# What can it do?

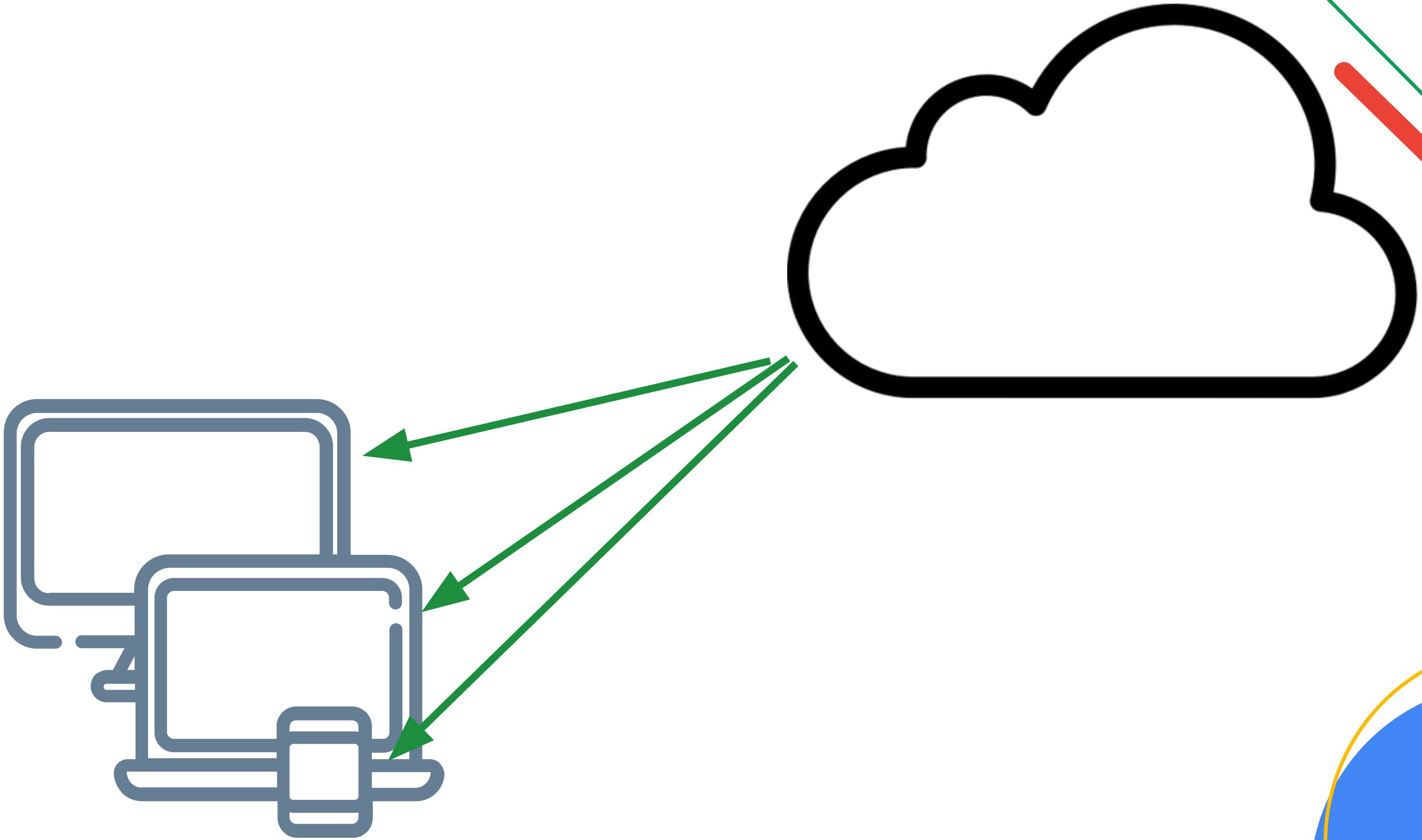


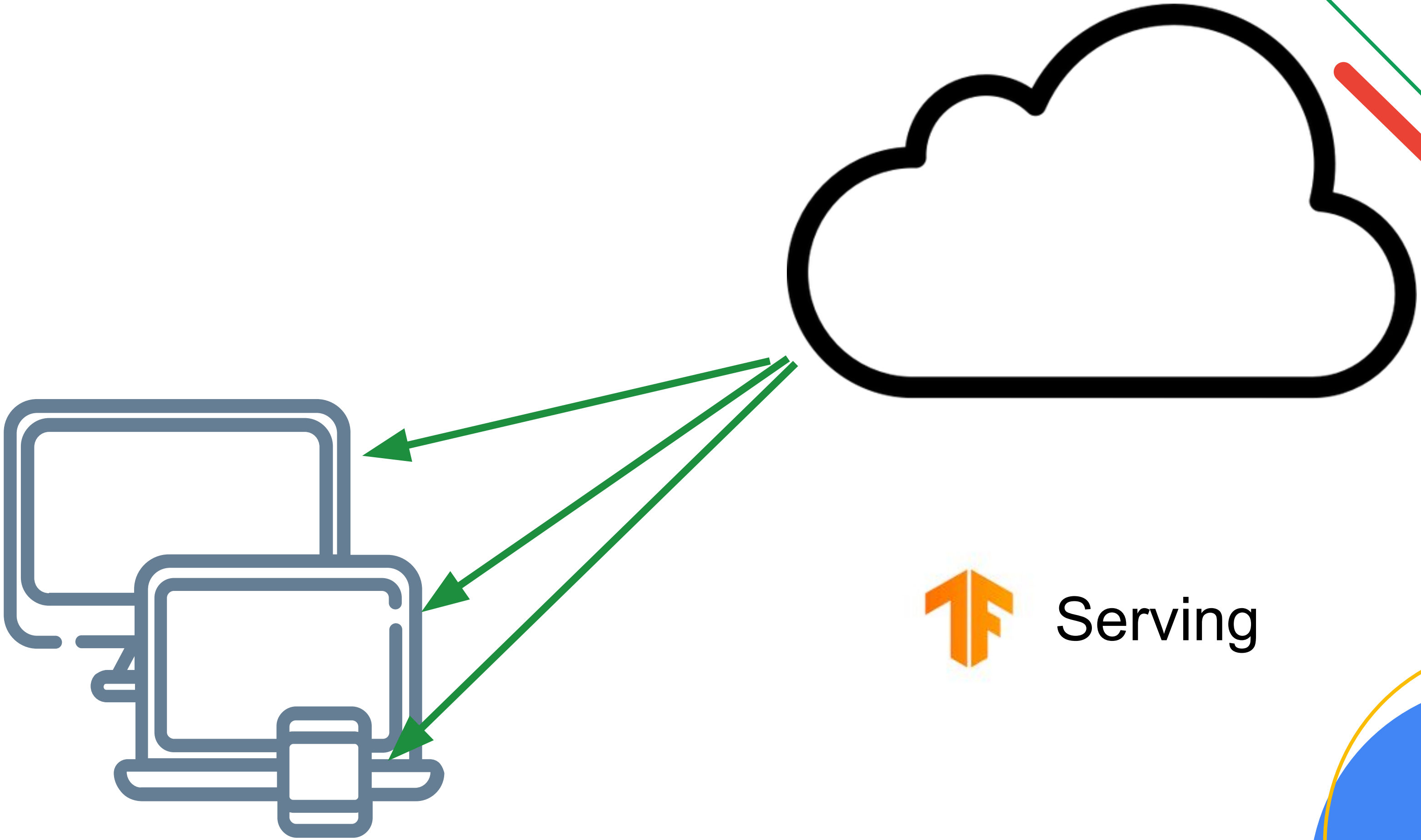












Serving

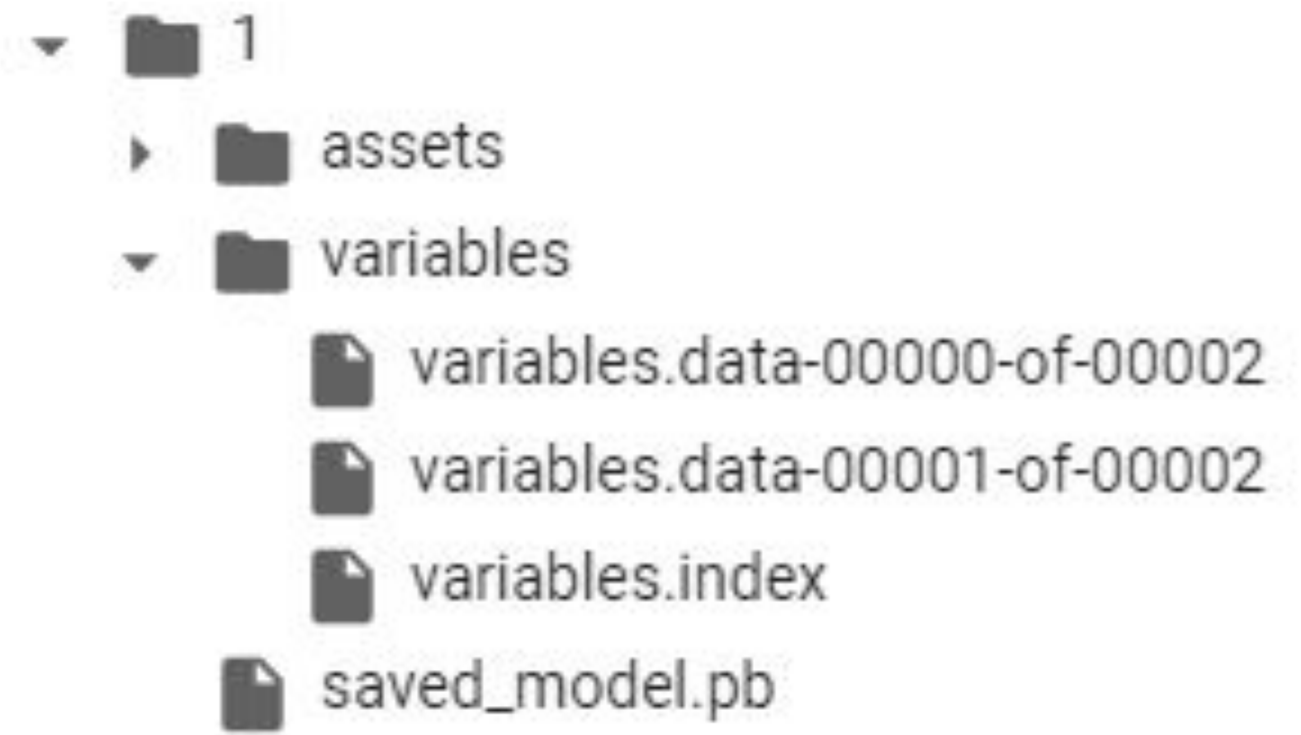
# Installing

https://www.tensorflow.org/tfx/serving/setup

# The Process

# Converting the model

```
tf.saved_model.simple_save(
    keras.backend.get_session(),
    directory_path,
    inputs = {'input_image': model.input},
    outputs = {i.name: i for i in model.outputs}
)
```



```
1
  assets
  variables
    variables.data-00000-of-00002
    variables.data-00001-of-00002
    variables.index
  saved_model.pb
```

# Starting the Model Server

```
os.environ["MODEL_DIR"] = MODEL_DIR
```

# Starting the model server

```
os.environ["MODEL_DIR"] = MODEL_DIR

%%bash --bg
nohup tensorflow_model_server \
  --rest_api_port = 8501 \
  --model_name = test \
  --model_base_path="${MODEL_DIR}" >server.log 2>&1
```

# Starting the model server

```
os.environ["MODEL_DIR"] = MODEL_DIR

%%bash --bg
nohup tensorflow_model_server \
  --rest_api_port = 8501 \
  --model_name = test \
  --model_base_path="${MODEL_DIR}" >server.log 2>&1
```

# Starting the model server

```
os.environ["MODEL_DIR"] = MODEL_DIR

%%bash --bg
nohup tensorflow_model_server \
  --rest_api_port = 8501 \
  --model_name = test \
  --model_base_path="${MODEL_DIR}" >server.log 2>&1
```

# Starting the model server

```
os.environ["MODEL_DIR"] = MODEL_DIR

%%bash --bg
nohup tensorflow_model_server \
  --rest_api_port = 8501 \
  --model_name = test \
  --model_base_path="${MODEL_DIR}" >server.log 2>&1
```

# Starting the model server

```
os.environ["MODEL_DIR"] = MODEL_DIR

%%bash --bg
nohup tensorflow_model_server \
  --rest_api_port = 8501 \
  --model_name = test \
  --model_base_path="${MODEL_DIR}" >server.log 2>&1
```

# Doing Inference!

# Keep In Mind

- No data as lists but as lists of lists

# Data as lists of lists

```
xs = np.array([[case_1], [case_2] ... [case_n]])
```

# Making calls

```
xs = np.array([[case_1], [case_2] ... [case_n]])

data = json.dumps({"signature_name": " ",
                   "instances": xs.tolist()})
```

# Doing Inference

```
xs = np.array([[case_1], [case_2] ... [case_n]])
data = json.dumps({"signature_name": " ",
                   "instances": xs.tolist()})


json_response = requests.post(
'http://localhost:8501/v1/models/test:predict',
data = data,
headers = headers)
```

# Doing Inference

```
xs = np.array([[case_1], [case_2] ... [case_n]])
data = json.dumps({"signature_name": " ",
                   "instances": xs.tolist()})

json_response = requests.post(
'http://localhost:8501/v1/models/test:predict',
data = data,
headers = headers)
```

# Doing Inference

```
xs = np.array([[case_1], [case_2] ... [case_n]])
data = json.dumps({"signature_name": " ",
                   "instances": xs.tolist()})


json_response = requests.post(
'http://localhost:8501/v1/models/test:predict',
data = data,
headers = headers)
```

# Doing Inference

```
xs = np.array([[case_1], [case_2] ... [case_n]])
data = json.dumps({"signature_name": " ",
                   "instances": xs.tolist()})

json_response = requests.post(
'http://localhost:8501/v1/models/test:predict',
data = data,
headers = headers)
```

- High availability

- High availability
- No downtime

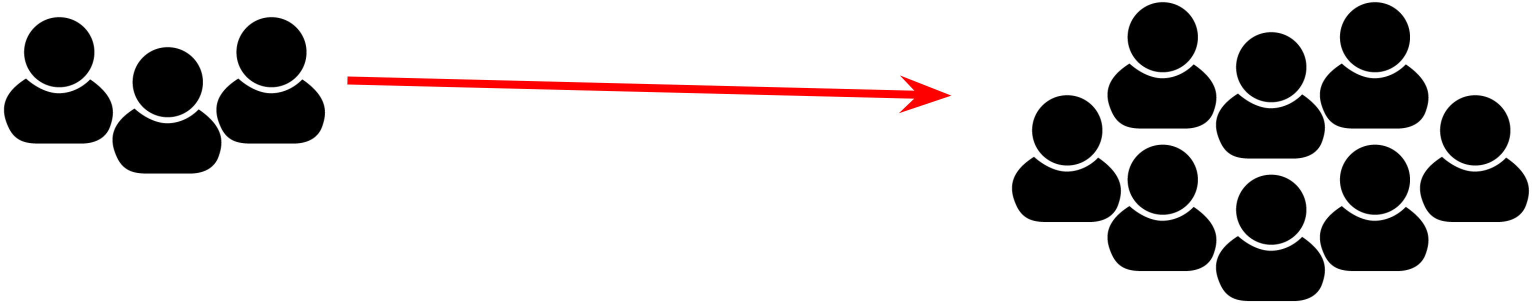
- High availability
- No downtime
- Focus on real code

- High availability
- No downtime
- Focus on real code
- Build better apps

# Serve the model on Cloud

# Why Care?

# Why Care?



# Creating a Kubernetes cluster

```
gcloud container clusters create
    resnet-serving-cluster
    --num-nodes 5
```

# Pushing the docker image

```
docker tag
    $USER/resnet_serving
    gcr.io/tensorflow-serving/resnet

docker push
      gcr.io/tensorflow-serving/resnet
```

# Pushing the docker image

kubectl create -f [yaml]

# Inference

- Use the external IP

# gdg-ahm.rishit.tech

Code Repo

# Demos

# Key Takeaways

- Why a process for deployment
- What it takes to deploy models
- Serving a model with TF Model server
- Why TF Model server?
- What can TF Model server do?
- Deploying on Cloud

# About Me

Rishit Dagli

Rishit-dagli

rishit_dagli

rishit.tech

hello@rishit.tech

@rishit.dagli

# Questions

# Thank You