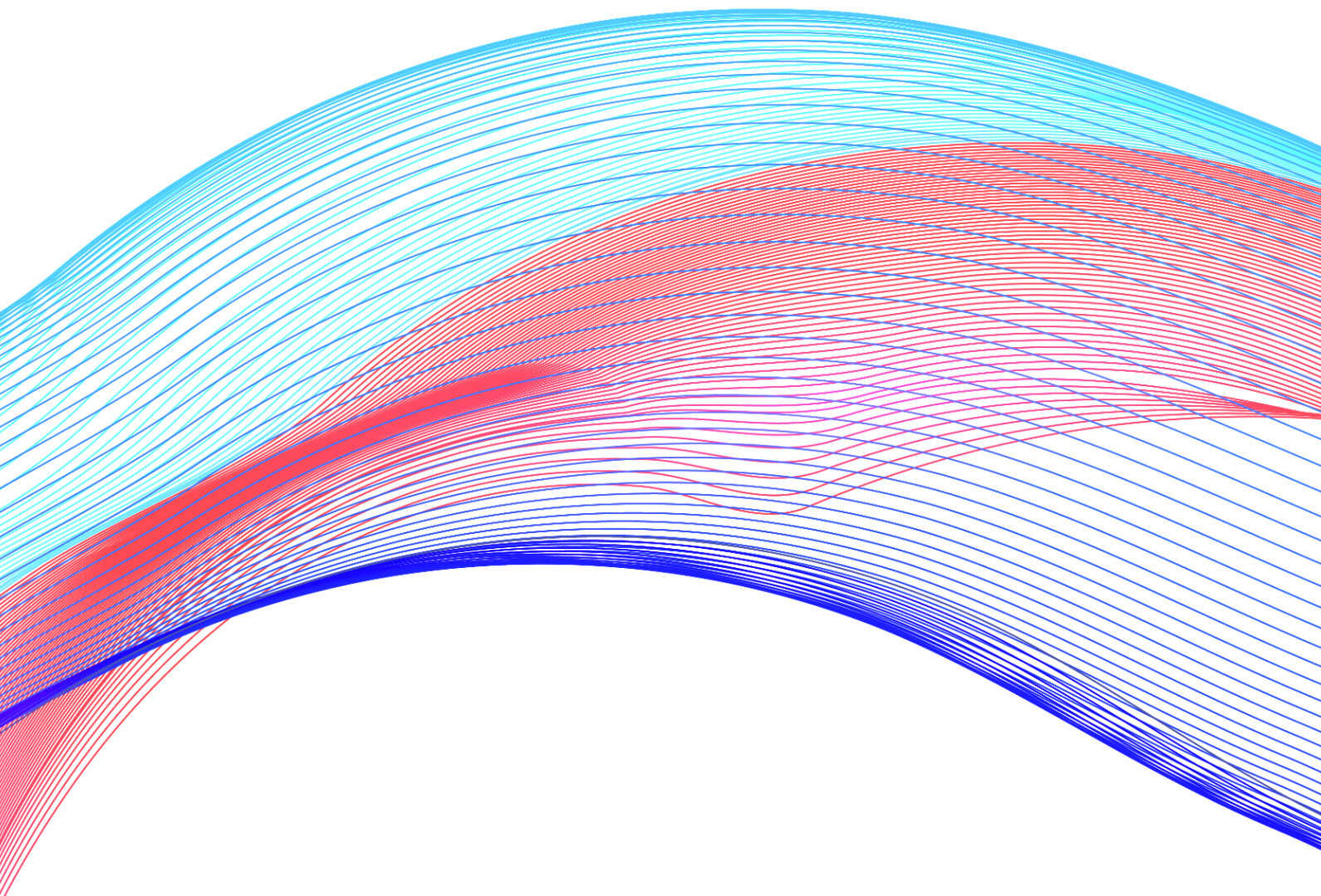




Image API 1.0.1

Administrator Guide



Contents

1. System Requirements	3
1.1. Minimum System Requirements	3
2. Deployment Instructions	4
2.1. Deployment Preparation	4
2.1.1 Preparation	4
2.1.2 Configuration	4
2.1.3 Installing Dependencies	4
2.1.4 Uploading Images	5
2.1.5 Installing and Configuring a Cluster	5
2.1.6 Cluster Health Check	5
2.2. Deployment	6
2.2.1 Configuration File Settings (optionally)	6
2.2.2 Installation	7
2.2.3 Scalability	8
2.2.4 Configure DNS	9
2.2.5 Testing	9
2.2.6 Deletion	10
2.3 Licensing	10
2.3.1 Install a license	10
2.3.1 Update a license	10
3. Access to Image API	11

Scope of delivery

Image API is delivered in the distribution kit. The distribution kit contains the files necessary for Image API deployment (setup directory) and documentation in pdf format. The link to download the distribution kit and the attached license file are sent to the client by e-mail.

1. System Requirements

1.1. Minimum System Requirements

Hardware:

CPU

- 6 cores (12 threads)
- 2.8GHz
- AVX/AVX2 support

RAM:

- 6GB (when launching one instance of each service)

SSD / HDD:

- 30+ GB free space
- 20%+ available space in the file system

The software below should be installed on your machine:

- OS Ubuntu 22.04
- Docker 20.10.17
- Kubernetes 1.23.8
- Helm 3.10.2

Note: To install Docker, Kubernetes and Helm, use the command from para. 2.1.3 (the Internet connection is required).

2. Deployment Instructions

2.1. Deployment Preparation

2.1.1 Preparation

Replace a license file `face_sdk.lic` to the `setup` folder of the unpacked distribution kit.

Further commands should be performed relative to the `setup` directory.

2.1.2 Configuration

The distribution kit contains the following configuration files:

- `./deployment.cfg` is a configuration file of Image API instance deployment.
- `./dependencies.cfg` is a configuration file of dependencies for installation of Kubernetes cluster.
- `./cluster.cfg` is a configuration file of Kubernetes cluster.

Open a deployment configuration file `./cluster.cfg` in a text editor and set values for the following variables:

- `apiserver_advertise_address` is the internal IP address of the machine for deployment.
- `external_ip_address` is the public IP address of the machine which DNS will be configured to.

Note: if there is no public IP address, use the internal IP address.

Open a deployment configuration file `./deployment.cfg` in a text editor and set values for the following variables:

- `domain` is a domain name.

IP address for domain name `<domain>` should be configured at DNS server (For more details see section 2.2.4).

2.1.3 Installing Dependencies

If the system has no Docker, Kubernetes or Helm, run the command below:

```
$ ./cli/dependencies/install.sh
```

2.1.4 Uploading Images

Upload docker-images to the local `registry`:

```
$ ./cli/load-images.sh
```

Uploading can last for about 5 minutes.

2.1.5 Installing and Configuring a Cluster

To create and configure the cluster, run the following command:

```
$ ./cli/cluster/install.sh
```

This command initializes a node for cluster deployment, configures the network, installs an ingress-controller and creates Kubernetes secrets.

2.1.6 Cluster Health Check

After initializing the master node, make sure that all nodes are ready for operation. You can check this by running the command below:

```
$ kubectl get nodes
```

As a result, the following output will be displayed in the terminal:

NAME	STATUS	ROLES	AGE	VERSION
master-node	Ready	control-plane,master	1d	v1.23.8

2.2. Deployment

2.2.1 Configuration File Settings (optionally)

Optionally, before installing Image API to a cluster, you can set values for the following fields in the `deployment.cfg` configuration file:

- **image_pull_policy:** the parameter affects pulling (downloading) container images from `registry` to deploy an Image API deployment instance. The following field values are available:
 - `Always` - kubelet (node agent in Kubernetes) pulls an image from `registry`;
 - `Never` - image pulling from `registry` is not applicable. If the image is already present locally, kubelet attempts to launch a container, otherwise, startup fails;
 - `IfNotPresent` - image is pulled only if it isn't present locally.
- **capturer_config:** the field value indicates the name of the configuration file of the selected face detector.

Image API uses face detectors from 3DiVi Face SDK - a set of code libraries for development of face recognition solutions.

Image API version 1.0.0 works with two detectors: ULD and REFA. ULD detector allows you to detect faces of different sizes, including masked faces (detection speed is higher than that of REFA). REFA detector provides higher coverage of head rotation angles and maximum quality, including detection of masked faces (detection quality is higher than that of ULD).

The following detector configuration files can be specified as values for the `capturer_config` field:

- `common_capturer_uld_fda.xml` (Configuration file for ULD detector)
- `common_capturer_refa_fda_a.xml` (Configuration file for REFA detector)

For more information about face detector configuration files, see the documentation of [3DiVi Face SDK](#).

- **recognizer_config:** the field value indicates the name of the face recognizer configuration file. Image API uses a set of recognizers from 3DiVi Face SDK. Each recognizer uses a specific identification method (algorithm). Generally, 3DiVi Face SDK provides several identification methods that differ in identification quality and operating time.

Image API 1.0.1 Administrator Guide

For Image API 1.0.0, you can specify the following recognizer configuration files:

- method11v1000_recognizer.xml
- method12v30_recognizer.xml
- method12v50_recognizer.xml
- method12v100_recognizer.xml
- method12v1000_recognizer.xml

For more information about recognizers, identification methods and recognizer configuration files, see [3DiVi Face SDK](#) documentation.

- **enable_use_avx2**: using the AVX2 instructions allows to speed up the recognizers. To enable/disable AVX2, specify the following values for the field: “1” (enabled) or “0” (disabled).
- **downscale_rawsamples**: field values: “1” (enabled) or “0” (disabled). By default the value is “0”. When enabled, each sample is downscaled to the suitable size to reduce memory consumption. However, it decreases the performance. It is recommended to leave the field disabled.

2.2.2 Installation

Launch the script below to install Image API to the cluster:

```
$ ./cli/deployment/install.sh
```

Then run the following command to get the deployment status:

```
$ kubectl get pods
```

As a result, the console will display a list of pods (deployment instances), their statuses, the number of restarts, and the pod age.

The example of console output:

NAME	READY	STATUS	RESTARTS	AGE
image-api-v1-0-0-age-estimator-dep-5bf7889dcd-jm46h	1/1	Running	0	1h
image-api-v1-0-0-body-detector-dep-6778495696-4zkpm	1/1	Running	0	1h
image-api-v1-0-0-emotion-estimator-dep-7c8cdd7c6b-z4v1v	1/1	Running	0	1h
image-api-v1-0-0-face-detector-face-fitter-dep-5fb48dcb5...	1/1	Running	0	1h
image-api-v1-0-0-gateway-dep-65d4846c59-g2nfj	1/1	Running	0	1h
image-api-v1-0-0-gender-estimator-dep-86b9457f9b-2464q	1/1	Running	0	1h
image-api-v1-0-0-face-detector-liveness-estimator-dep-7bd...	1/1	Running	0	1h
image-api-v1-0-0-mask-estimator-dep-76b958b84b-w7pss	1/1	Running	0	1h
image-api-v1-0-0-quality-assessment-estimator-dep-fbf8bc4...	1/1	Running	0	1h
image-api-v1-0-0-face-detector-template-extractor-dep-5758...	1/1	Running	0	1h
image-api-v1-0-0-verify-matcher-dep-6d66dc4948-1x8j7	1/1	Running	0	1h

Note: Status `Running` for all pods means that all containers are successfully launched.

Overview of the deployments is given below:

- `image-api-v1-0-0-gateway-dep`: reverse proxy, responsible for unified access to processing services;
- `image-api-v1-0-0-face-detector-face-fitter-dep`: face detection and calculation of anthropometric points and head rotation angles (roll, pitch, yaw);
- `image-api-v1-0-0-body-detector-dep`: body detection;
- `image-api-v1-0-0-emotion-estimator-dep`: estimation of emotions;
- `image-api-v1-0-0-age-estimator-dep`: age estimation;
- `image-api-v1-0-0-gender-estimator-dep`: gender estimation;
- `image-api-v1-0-0-face-detector-liveness-estimator-dep`: the service checks if a person in the image is real or fake;
- `image-api-v1-0-0-mask-estimator-dep`: the service checks if a person in the image is wearing a medical face mask;
- `image-api-v1-0-0-quality-assessment-estimator-dep`: image quality assessment;
- `image-api-v1-0-0-verify-matcher-dep`: the service compares two biometric templates to verify whether they belong to the same person;
- `image-api-v1-0-0-template-extractor-dep`: the service extracts a biometric template from face image.

2.2.3 Scalability

When the load increases, for stable operation of Image API you can manually scale the deployments described in section 2.2.2.

To scale the installed release, run the command below:

```
$ kubectl scale deployment <deployment_name> --replicas <count>
```

where `<deployment_name>` is a deployment name (for example, `face-detector-liveness-estimator-dep`), and `<count>` is a number of deployment replicas (pods).

To save the scaling state, set the value for **replicas** key in `setup/chart/processing-services.json` file and restart the deployment using a command below:

```
$ ./cli/deployment/install.sh
```

2.2.4 Configure DNS

To allow DNS access, your network server should contain a record that `<domain>` is available at `<external_ip_address>`. Variable values can be obtained from the `./dependencies.cfg` and `./cluster.cfg` files filled in in section 2.1.2. To perform this configuration, contact your network administrator.

For testing you can specify the IP address and domain in the `/etc/hosts` file on Linux or `C:\Windows\System32\drivers\etc\hosts` on Windows. To do this, add a new line `<external_ip_address> <domain>` with the values of the corresponding variables to the end of this file and save the file. Please note that you must have administrator rights to edit the `hosts` file.

To use Image API at the same machine where it is deployed, you can use a script that will automatically add the necessary record to the `/etc/hosts` file.

```
$ ./cli/add-dns.sh
```

2.2.5 Testing

To make sure that the system works properly, use the command below:

```
$ ./cli/test.sh
```

Test output should be the following:

```
face-detector-face-fitter ✓  
age-estimator ✓  
emotion-estimator ✓  
gender-estimator ✓  
face-detector-liveness-estimator ✓  
mask-estimator ✓  
quality-assessment-estimator ✓  
face-detector-template-extractor ✓  
body-detector ✓
```

```
face-detector-template-extractor ✓  
face-detector-template-extractor ✓  
verify-matcher ✓
```

2.2.6 Deletion

To delete Image API, run the command below:

```
$ ./cli/deployment/delete.sh
```

2.3 Licensing

2.3.1 Install a license

Unpack Image API distribution kit and move a license file `face_sdk.lic` (the file is attached to the e-mail) to `setup` directory.

2.3.2 Update a license

1. Delete an object of “secret” type, that contains a license file, from Kubernetes cluster.

```
$ kubectl delete secret face-sdk
```

2. Install a secret into the cluster, where `./face_sdk.lic` is a path to the license file.

```
$ face_sdk_lic_path=./face_sdk.lic ./cli/cluster/install/secrets.sh
```

3. Delete Image API release from the cluster.

```
$ ./cli/deployment/delete.sh
```

4. Install Image API release into the cluster.

```
$ ./cli/deployment/install.sh
```

3. Access to Image API

To review how Image API services work, we provide a special web interface available at **<image_api_url>**. From this entry point you can get access to the Swagger documentation.

<image_api_url> is represented as *http://<domain>*, where *<domain>* is a domain name, specified in para. 2.1.2.

To directly access the Swagger documentation for a specific service, go to **<image_api_url>/<service_name>/docs**, where *image_api_url* is the address of the deployed product, and *service_name* is the service name.

List of service names:

- face-detector-face-fitter
- face-detector-liveness-estimator
- face-detector-template-extractor
- age-estimator
- emotion-estimator
- gender-estimator
- mask-estimator
- quality-assessment-estimator
- body-detector
- verify-matcher

Note: If you have any questions, need installation assistance or want to leave your feedback, contact 3DiVi support team at the address support-image-api@3divi.com.