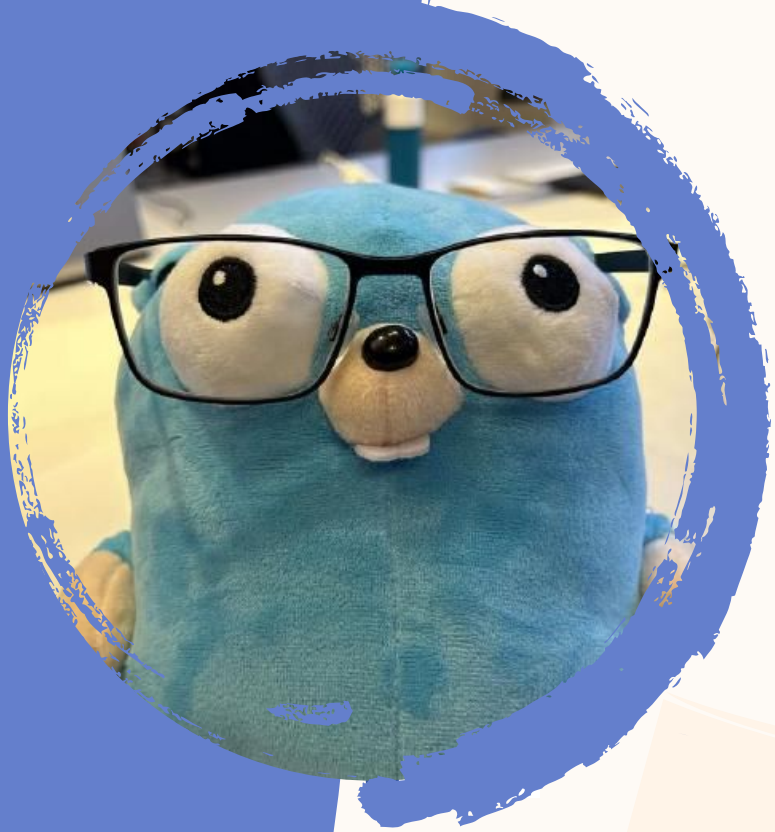
Creative Fabrica

Our Journey on Processing Digital Assets

PNGs
Sketches
Cartoons

Mockups
Patterns
Backgrounds

Stock Photos
Sketches
Cartoons



Herlon Aguiar 🇧🇷

Staff Software Engineer

@herlon214

E-Mail

herlon@creativefabrica.com

We are Creative Fabrica

We believe in empowering people, giving them the flexibility and freedom to do what's best, take on new challenges and create their own opportunities.

We only hire the best, an international and diverse team of highly talented and creative people.

We're ambitious, this is a fast scaling VC funded startup, and people can really make their mark.

Our platform

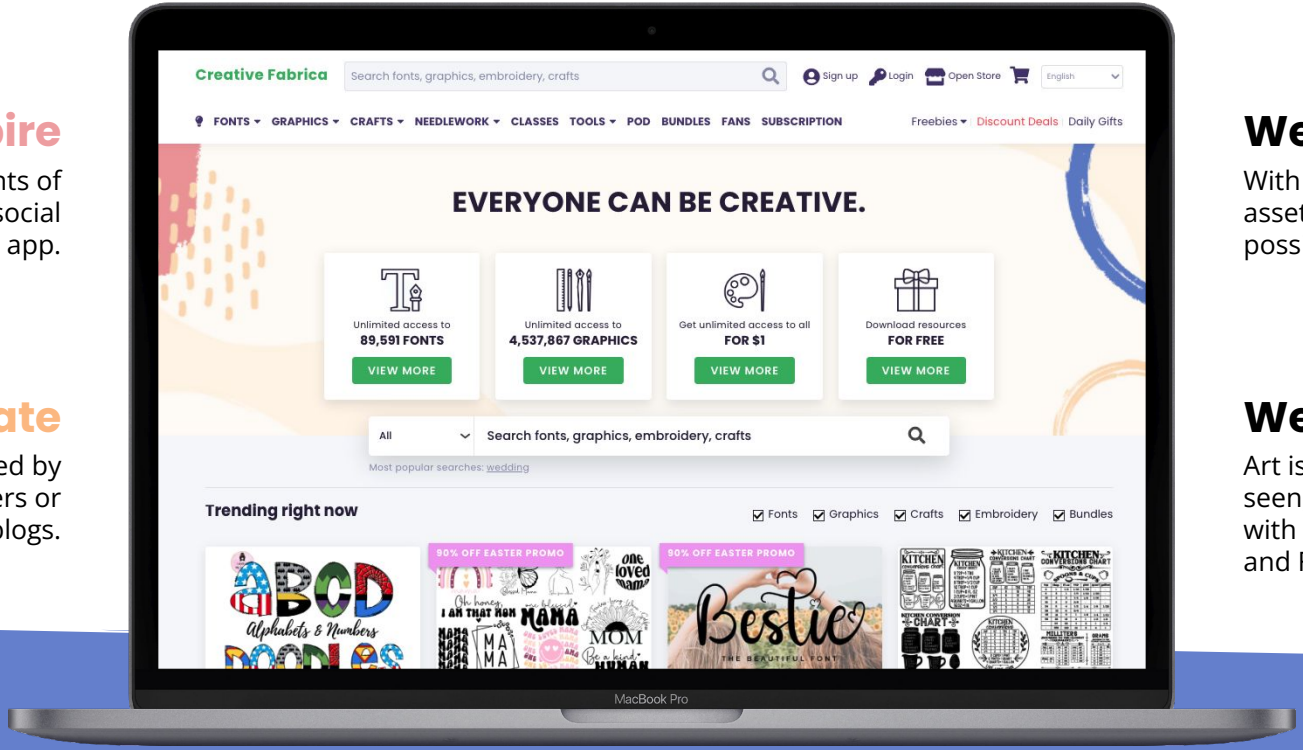
40 million+ monthly page views and growing 🚀

We inspire

With endless amounts of creative assets, social feed, blog and app.

We educate

Join classes hosted by your favorite crafters or read explanatory blogs.

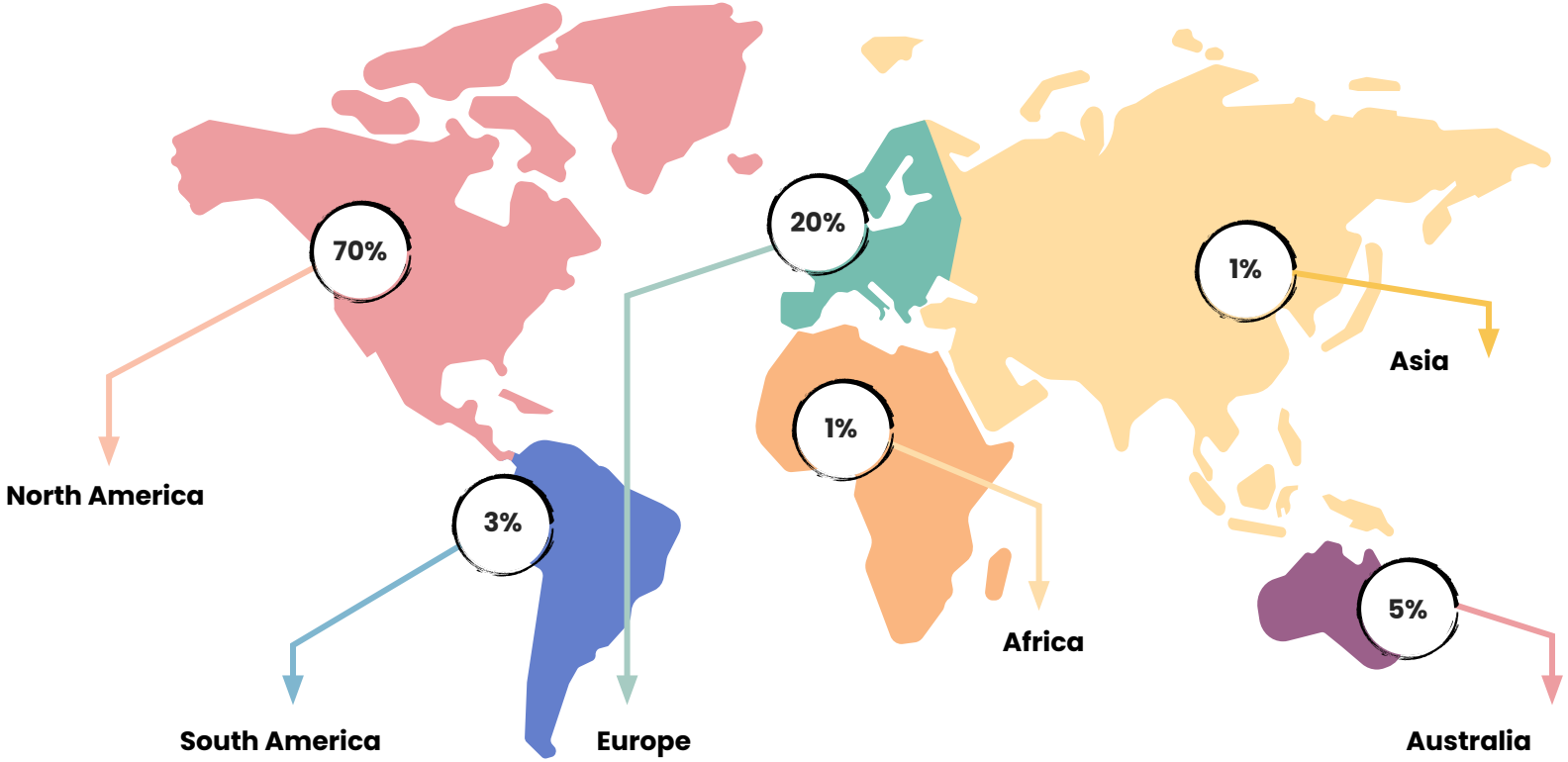


We create

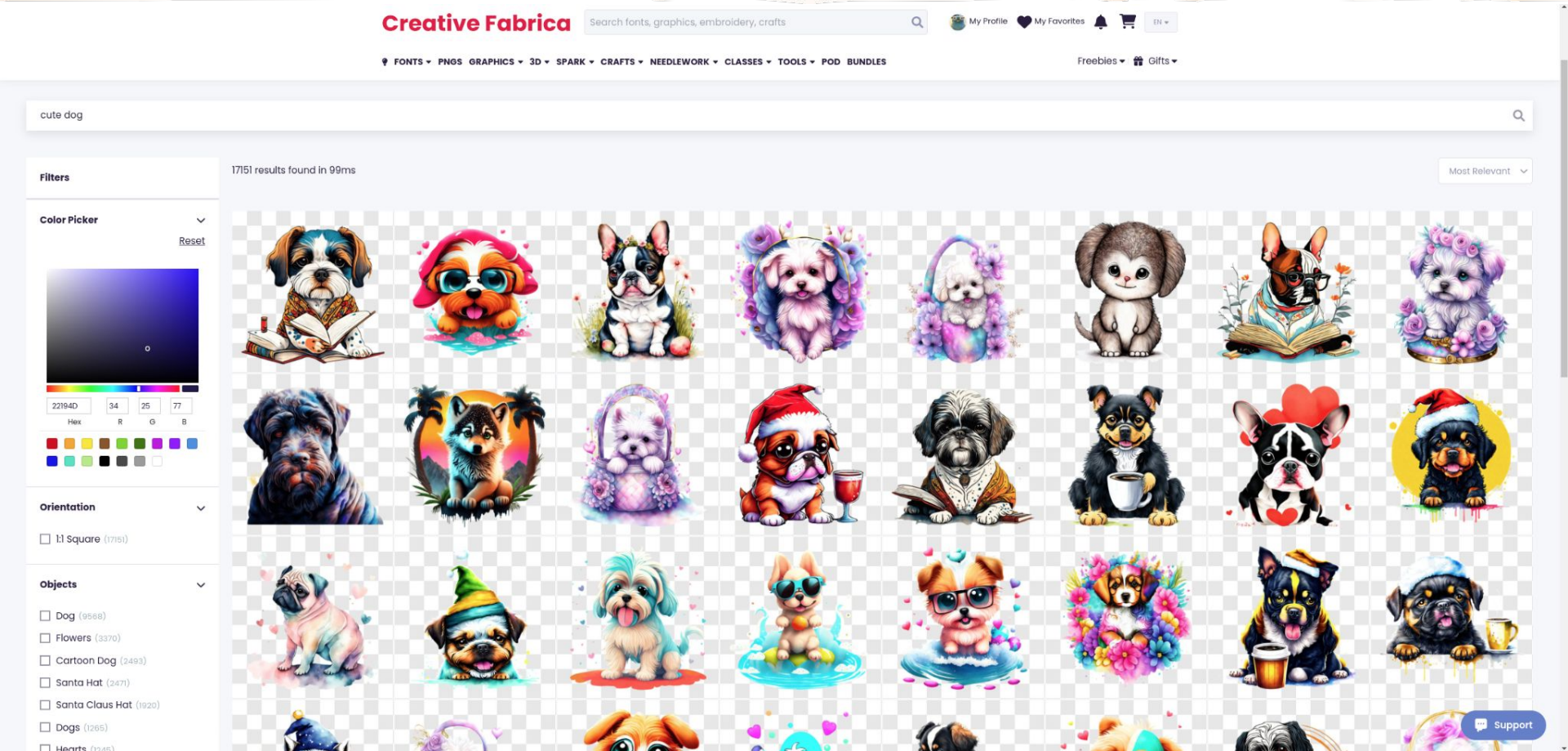
With our downloadable assets, any creation is possible.

We share

Art is to be shared and seen, which is possible with our social layer and Fans platform.

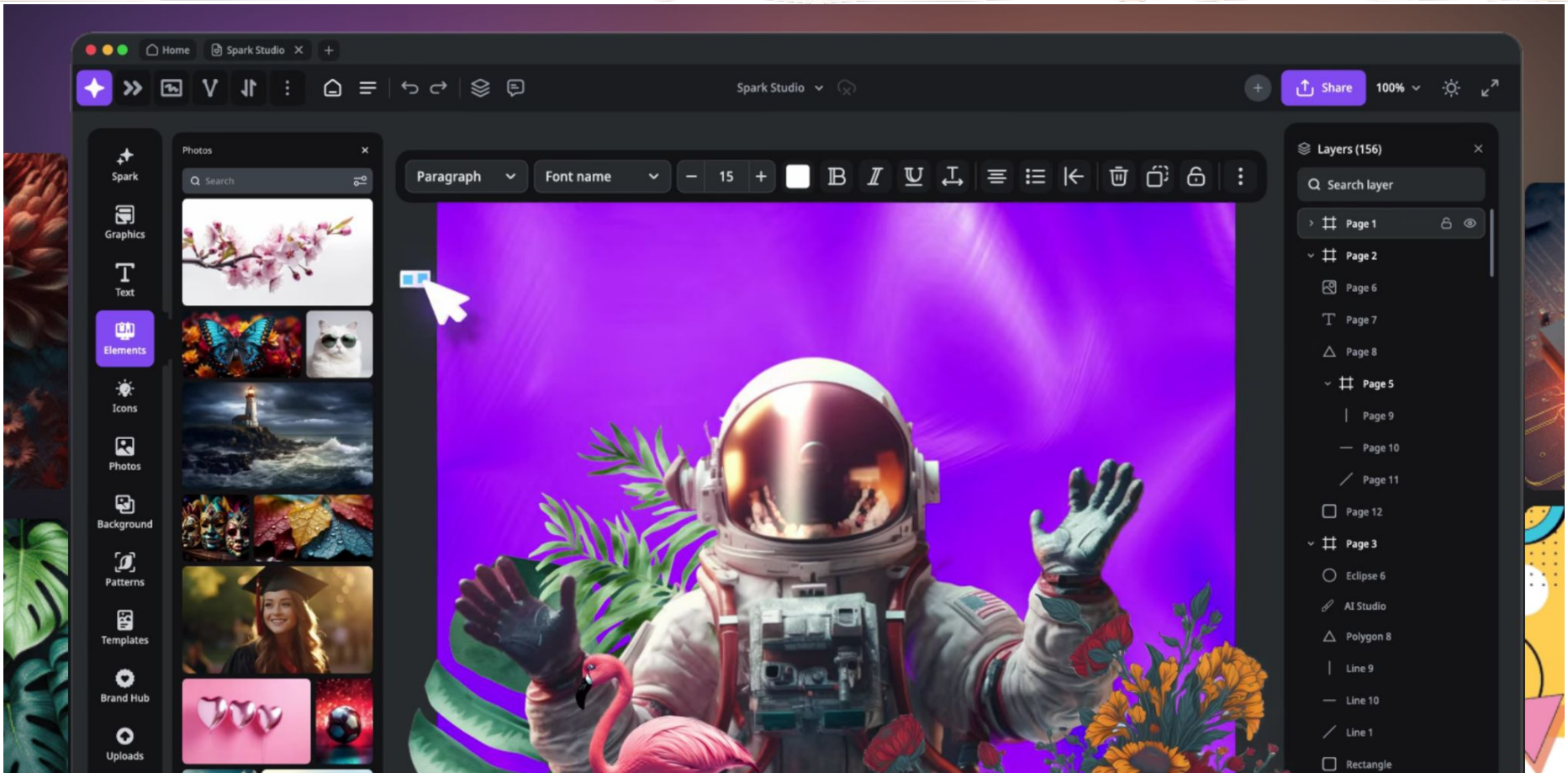

70%
North America
3%
South America
20%
Europe
1%
Africa
1%
Asia
5%
Australia

Example of products



What can I do with the products?

Studio (creativefabrica.com/studio)



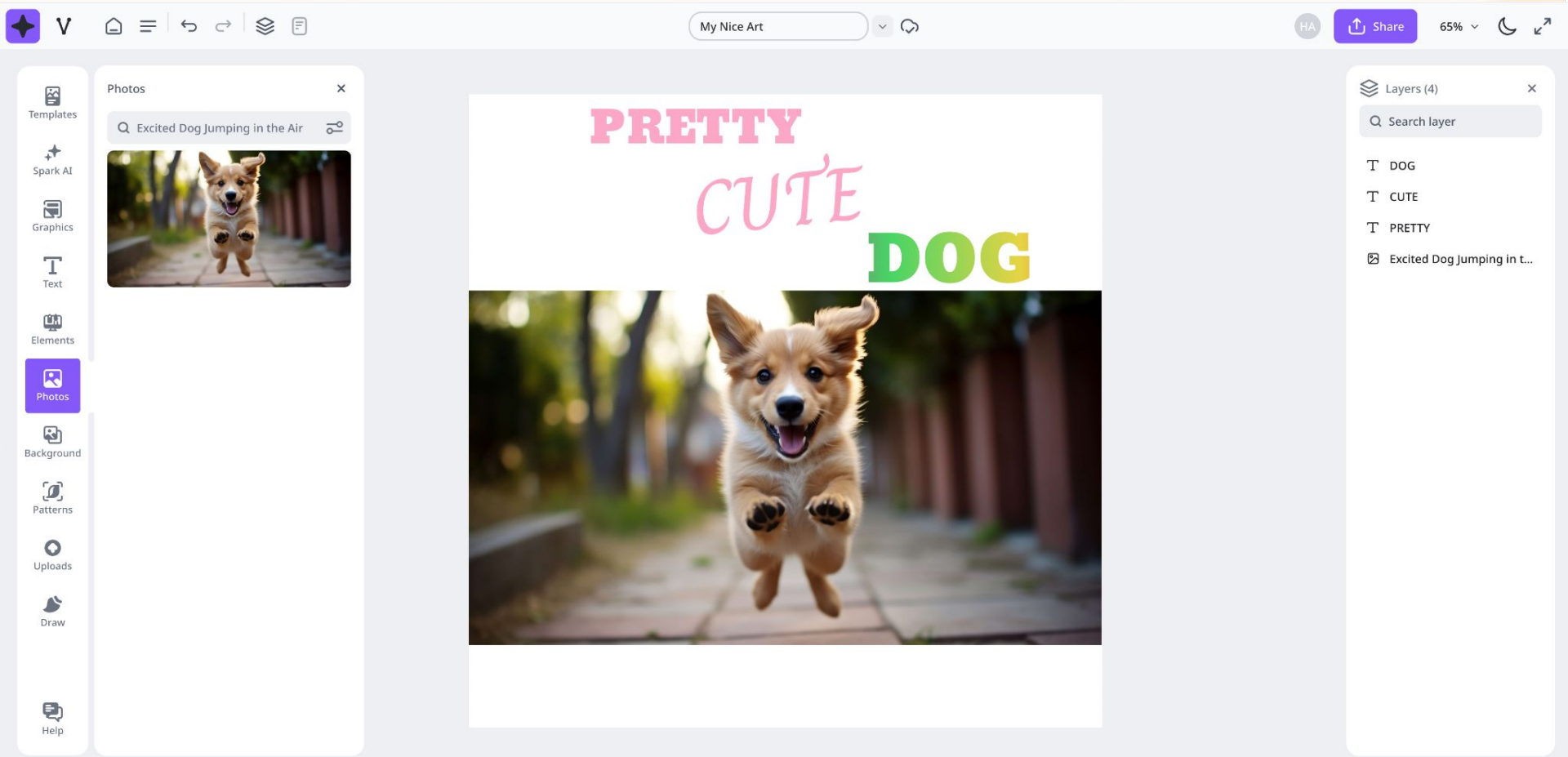
Another example

Creative Fabrica

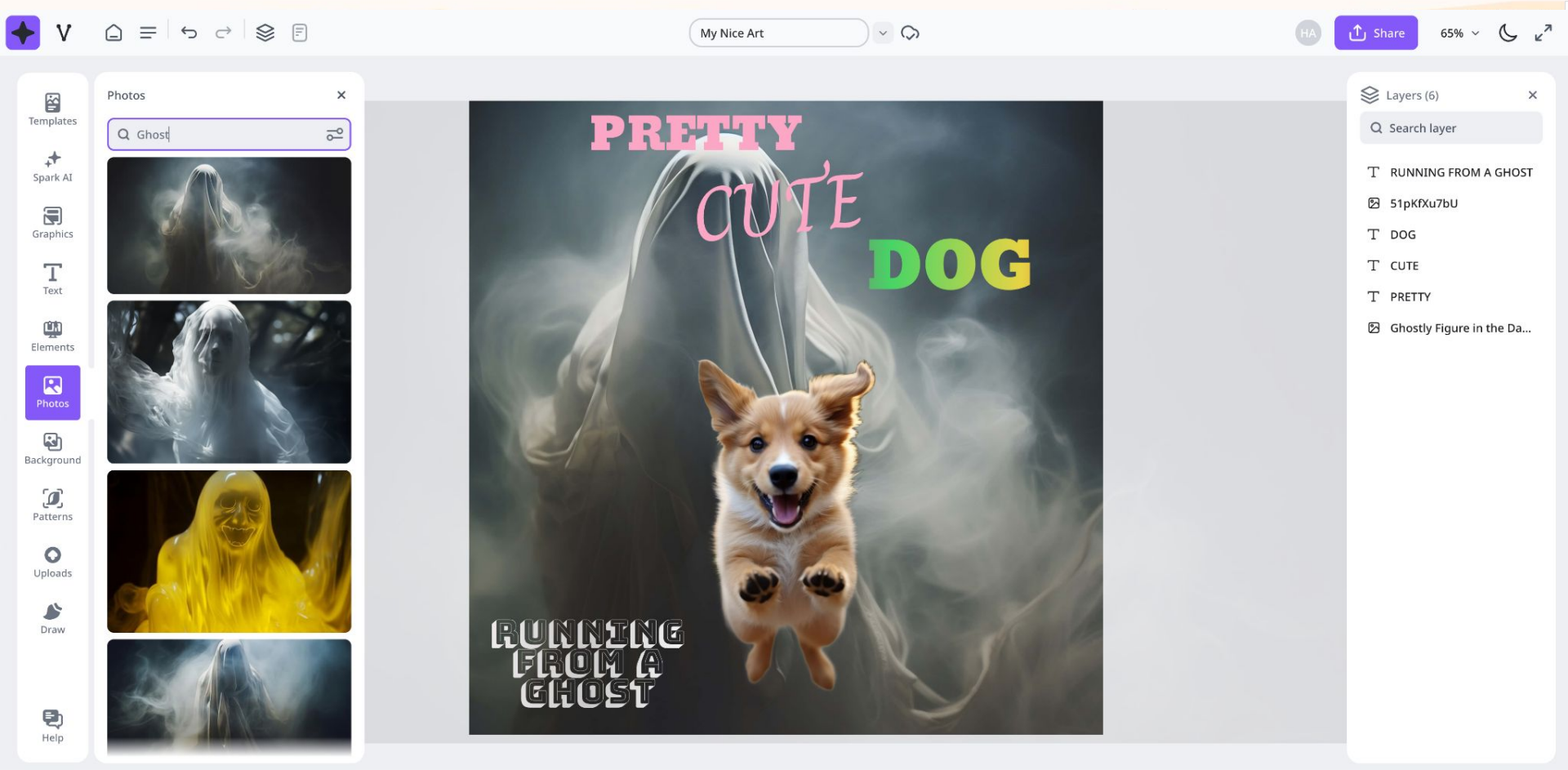
You're cordially invited
to the Go Meetup at
Creative Fabrica

Bring friends!

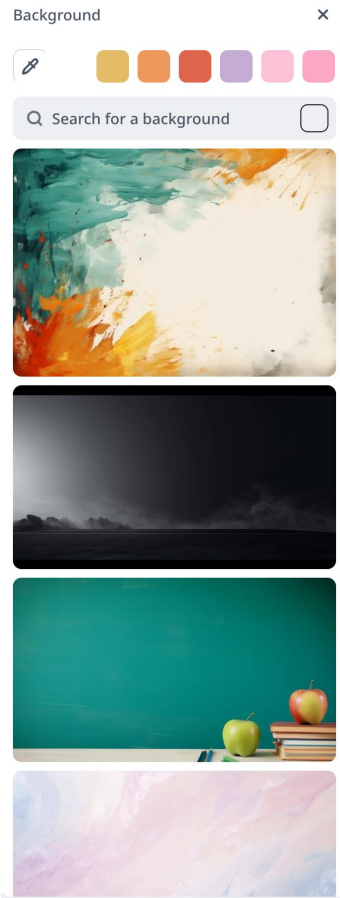
Use the assets on Studio



It gets even better!



More examples



Product Page

Home / Stock Photos / Animals / Excited Dog Jumping in the Air with Happiness

Excited Dog Jumping in the Air with Happiness

Share

Pin

Choose your size

- 640 X 359 (261 KB)
- 1024 X 574 (628 KB)
- 1920 X 1076 (2.09 MB)
- 4096 X 2296 (3.78 MB)

DOWNLOAD

69 19

All stock photos come with a royalty-free worldwide lifetime license.

SHARE & EARN

Share the link below and earn 25% for every purchase made by any customer using your link!

https://www.creativefabrica.com/prod/happy-dog-ju COPY

Listed on Sep 20, 2023 - ID ut4hrMwDo

Report problem

Dog | Jumping | Excited | Happy | Pet

About Excited Dog Jumping in the Air with Happiness

This image features a small brown and white dog jumping up in the air with its front paws extended and its tail wagging. The dog appears to be incredibly excited and happy. It's a heartwarming sight that showcases the joy and exuberance of our beloved pets.

EVERYTHING HERE IS BIASED

*Unless you think it's not**

Content Service

Tech used in the project

Kubernetes

Container orchestration platform that automates deployment, scaling, and management of containerized applications, optimizing resource usage and enhancing fault tolerance.

Kafka

Distributed streaming platform enabling high-throughput, fault-tolerant, and scalable real-time data processing.

Open Telemetry

Observability using tracing and metrics enhances application performance and reliability by identifying bottlenecks and failures in distributed systems.

Content Service

Tech used in the project

Kubernetes

Let you run containers and scale them easily.

Kafka

Reliable async communication.

Open Telemetry

Help you finding bottlenecks and see how things are going.

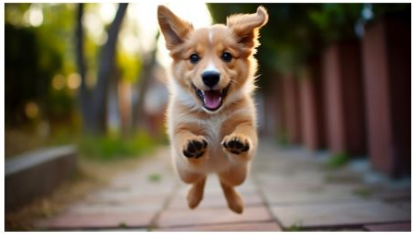
Content Service

Tech used in the project

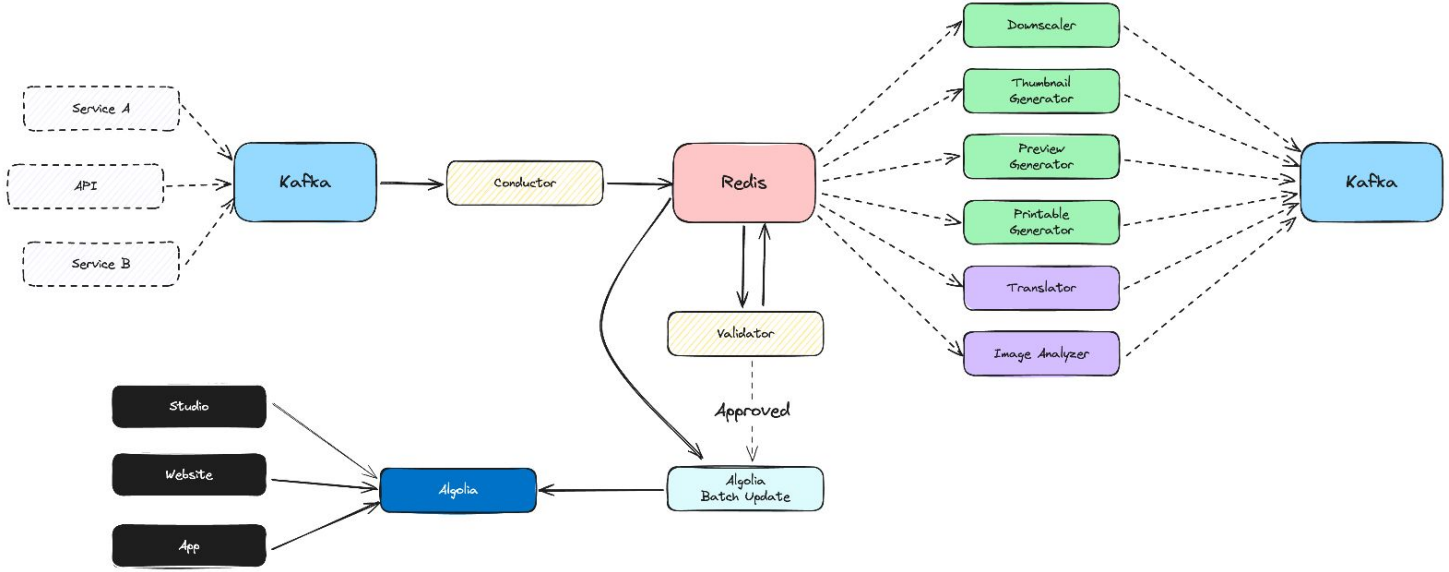
GO

You already know what it is.

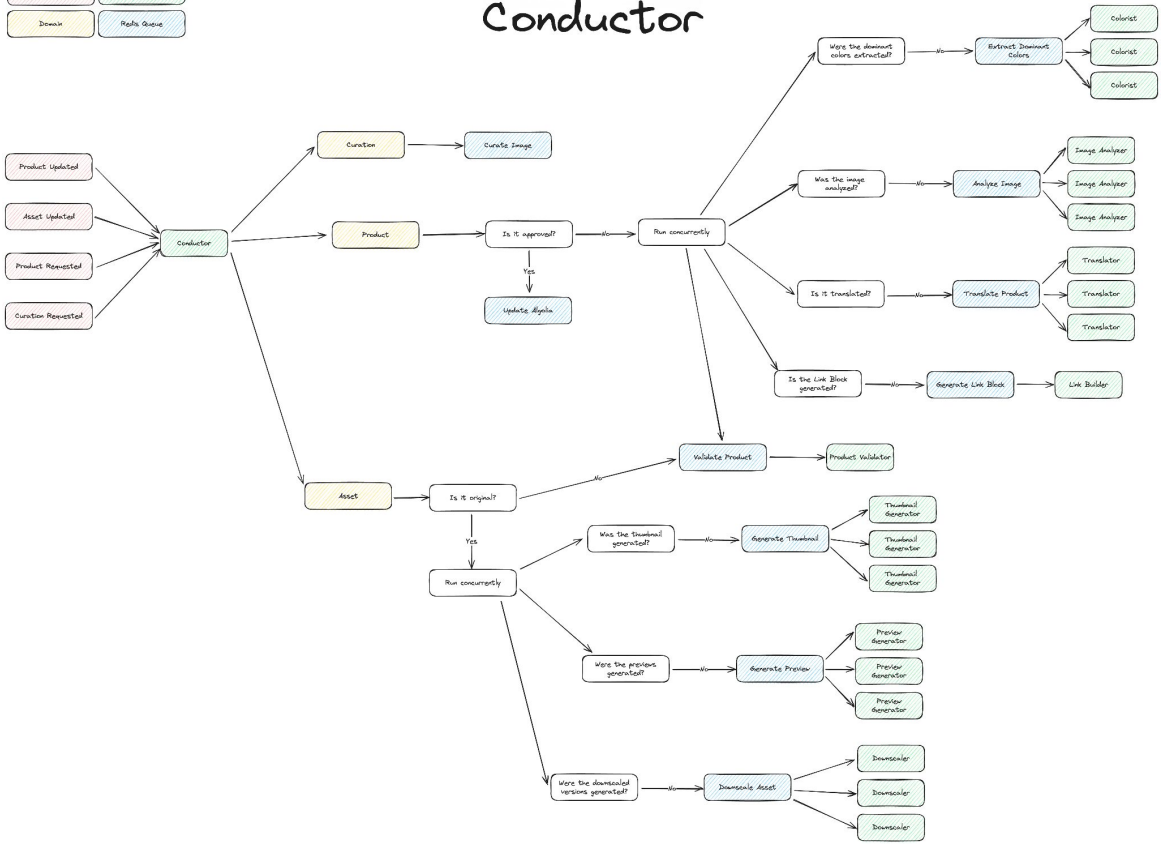
Processing Pipeline



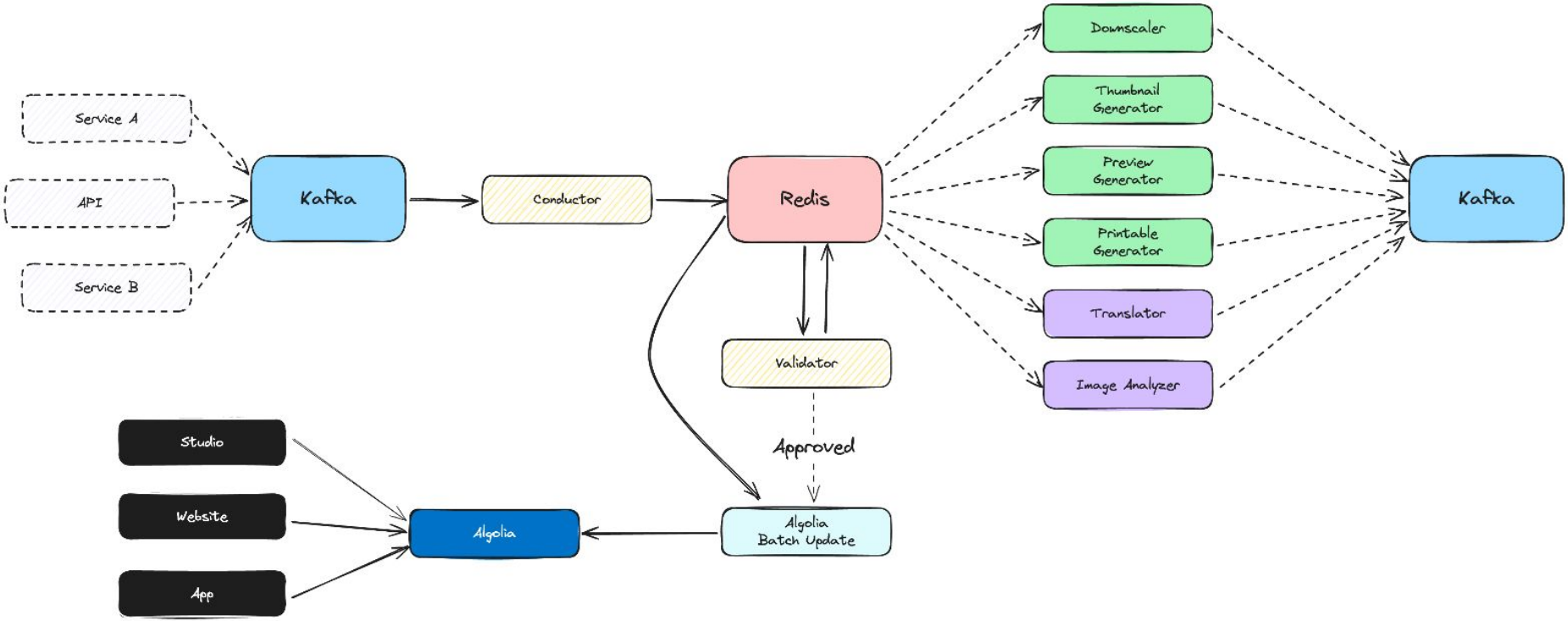
```
{
  "author": "stockphotos",
  "flags": {
    "enableDownscaling": true
    // other flags ...
  }
}
```



Conductor


Legend:
Kafka Topic
Service
Domain
Redis Queue
Product Updated
Asset Updated
Product Requested
Curation Requested
Conductor
Curation
Curate Image
Product
Is it approved?
No
Yes
Update Algolia
Run concurrently
Were the dominant colors extracted?
No
Extract Dominant Colors
Colorist
Colorist
Colorist
Was the image analyzed?
No
Analyze Image
Image Analyzer
Image Analyzer
Image Analyzer
Is it translated?
No
Translate Product
Translator
Translator
Translator
Is the Link Block generated?
No
Generate Link Block
Link Builder
Validate Product
Product Validator
Asset
Is it original?
No
Yes
Run concurrently
Was the thumbnail generated?
No
Generate Thumbnail
Thumbnail Generator
Thumbnail Generator
Thumbnail Generator
Were the previews generated?
No
Generate Preview
Preview Generator
Preview Generator
Preview Generator
Were the downscaled versions generated?
No
Downscale Asset
Downscaler
Downscaler
Downscaler

Processing Pipeline

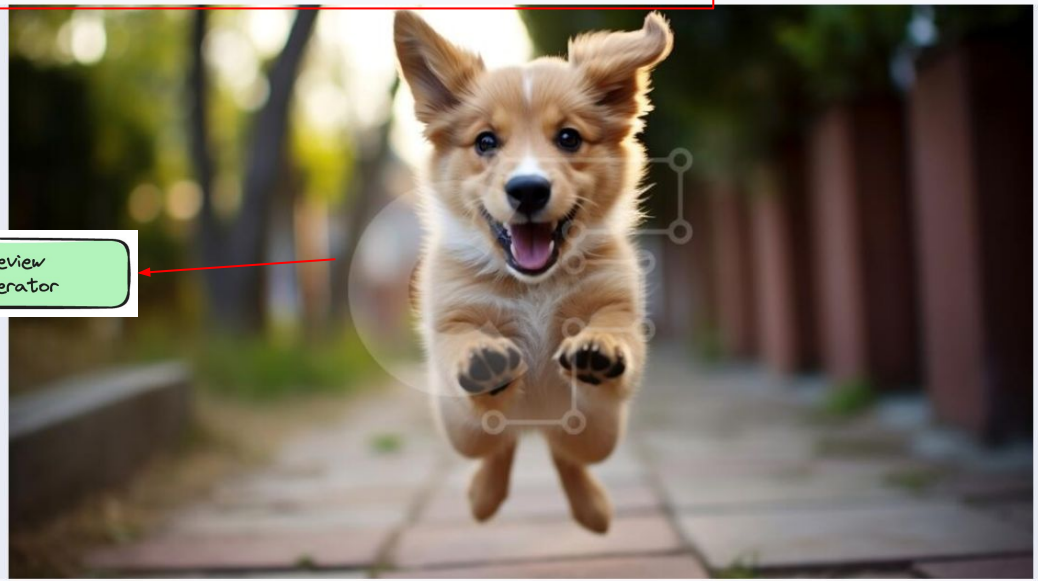


Product Page

Translator + Image Analyzer

Home / Stock Photos / Animals / Excited Dog Jumping in the Air with Happiness

Excited Dog Jumping in the Air with Happiness



Preview Generator

Share

Pin

Choose your size

- 640 X 359 (261 KB)
- 1024 X 574 (628 KB)
- 1920 X 1076 (2.09 MB)
- 4096 X 2296 (3.78 MB)

Downscaler

DOWNLOAD

69 19

All stock photos come with a royalty-free worldwide lifetime license.

SHARE & EARN

Share the link below and earn 25% for every purchase made by any customer using your link!

https://www.creativefabrica.com/prod/happy-dog-ju COPY

Listed on Sep 20, 2023 - ID ut4hrMwDo Report problem

Dog | Jumping | Excited | Happy | Pet

Translator + Image Analyzer

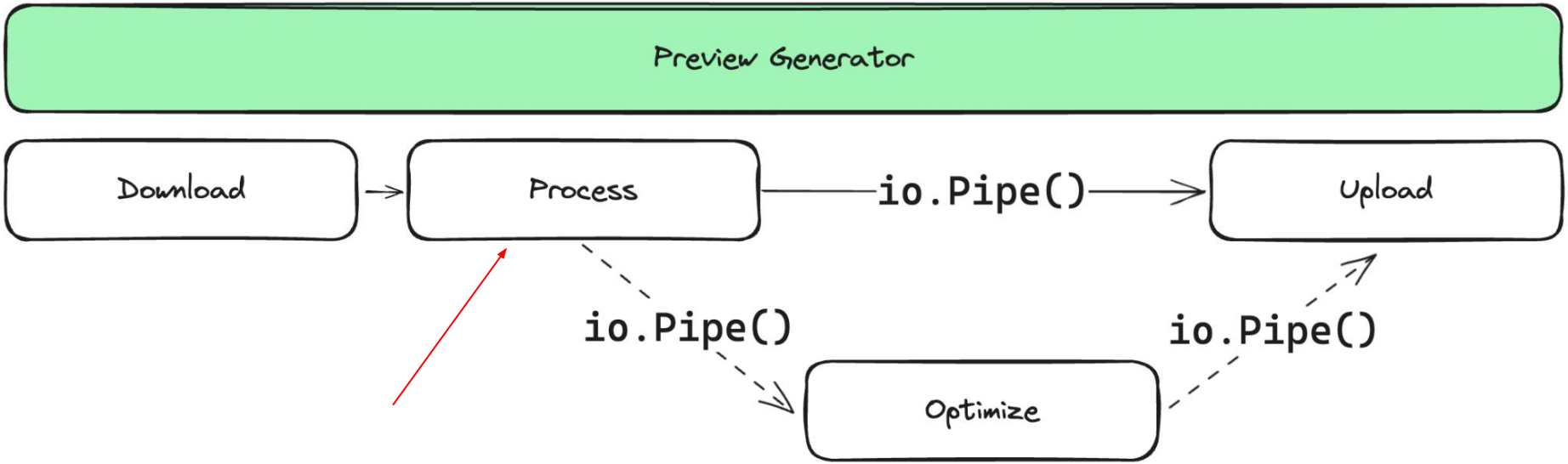
About Excited Dog Jumping in the Air with Happiness

This image features a small brown and white dog jumping up in the air with its front paws extended and its tail wagging. The dog appears to be incredibly excited and happy. It's a heartwarming sight that showcases the joy and exuberance of our beloved pets.

Preview Generator

Watermark

Processing Pipeline



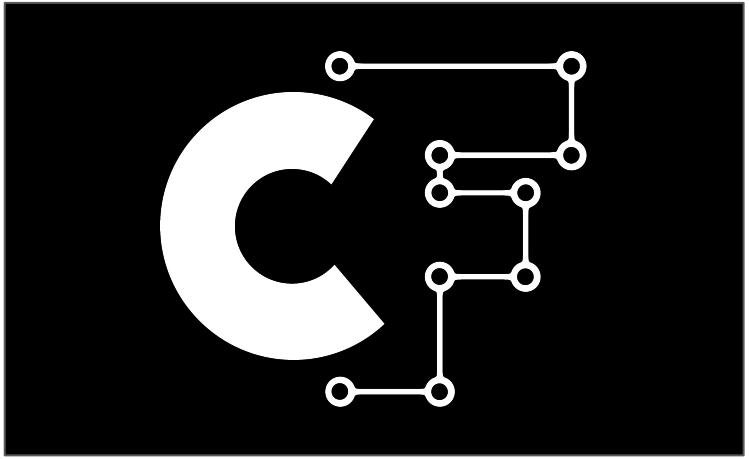
Watermark

Original Image

~6MB
5824x3264

Watermark

~600 KB
10000x8294



Watermark using GO

Dependencies

```go
import (
    "bytes"
    "fmt"
    "image"
    "image/color"
    "image/draw"
    "image/jpeg"
    "image/png"
    "os"
    "time"

    "github.com/anthonynsimon/bild/transform"
)
```

Watermark using GO

```go
func main() {
  for {
    startTime := time.Now()

    ghostFile, _ := os.Open("illustration.png")
    ghostImage, _ := png.Decode(ghostFile)

    result, err := GenerateWatermark(ghostImage, 0.5)
    if err != nil {
      panic(err)
    }

    // Encode image
    encodedImg := bytes.NewBuffer([]byte{})
    if err := jpeg.Encode(encodedImg, result, &jpeg.Options{Quality: 100}); err != nil {
      panic(err)
    }

    os.WriteFile("result.jpg", encodedImg.Bytes(), 0644)

    // Calculate elapsed time
    elapsedTime := time.Since(startTime)
    fmt.Println("Elapsed time:", elapsedTime)
  }
}
```

Watermark using GO

```go
func GenerateWatermark(origin image.Image, opacity float64) (image.Image, error) {
  // Prepare the watermark
  watermark, err := prepareWatermark(origin)
  if err != nil {
    return nil, err
  }

  // Calculate center point
  midX, midY := origin.Bounds().Dx()/2, origin.Bounds().Dy()/2
  wMidX, wMidY := watermark.Bounds().Dx()/2, watermark.Bounds().Dy()/2
  offset := image.Pt(midX-wMidX, midY-wMidY)

  // Create a new image
  newImg := image.NewRGBA(origin.Bounds())

  // Draw the origin image
  draw.Draw(newImg, origin.Bounds(), origin, image.Point{}, draw.Src)

  // Create the mask using the opacity
  rgba := color.RGBA{R: 0, G: 0, B: 0, A: uint8(opacity * 255)}
  mask := image.NewUniform(rgba)

  // Draw the watermark with the mask
  draw.DrawMask(newImg, watermark.Bounds().Add(offset), watermark, image.Point{}, mask, image.Point{}, draw.Over)

  return newImg, nil
}
```

Watermark using GO

```go
func prepareWatermark(origin image.Image) (*image.RGBA, error) {
  watermarkFile, err := os.Open("watermark.png")
  if err != nil {
    return nil, err
  }

  watermark, err := png.Decode(watermarkFile)
  if err != nil {
    return nil, err
  }

  halfHeight := origin.Bounds().Dy() / 2
  diff := watermark.Bounds().Dy() / halfHeight

  resized := transform.Resize(
    watermark,
    watermark.Bounds().Dx()/diff,
    watermark.Bounds().Dy()/diff,
    transform.Linear,
  )

  return resized, nil
}
```

Watermark using GO

```go
func GenerateWatermark(origin image.Image, opacity float64) (image.Image, error) {
  // Prepare the watermark
  watermark, err := prepareWatermark(origin)
  if err != nil {
    return nil, err
  }

  // Calculate center point
  midX, midY := origin.Bounds().Dx()/2, origin.Bounds().Dy()/2
  wMidX, wMidY := watermark.Bounds().Dx()/2, watermark.Bounds().Dy()/2
  offset := image.Pt(midX-wMidX, midY-wMidY)

  // Create a new image
  newImg := image.NewRGBA(origin.Bounds())

  // Draw the origin image
  draw.Draw(newImg, origin.Bounds(), origin, image.Point{}, draw.Src)

  // Create the mask using the opacity
  rgba := color.RGBA{R: 0, G: 0, B: 0, A: uint8(opacity * 255)}
  mask := image.NewUniform(rgba)

  // Draw the watermark with the mask
  draw.DrawMask(newImg, watermark.Bounds().Add(offset), watermark, image.Point{}, mask, image.Point{}, draw.Over)

  return newImg, nil
}
```

Watermark using ImageMagick

Dependencies

```
import (
    "fmt"
    "time"

    "github.com/creativefabrica/cf-go-lib/magick"
)
```

Watermark using ImageMagick

```go
func main() {
  for {
    startTime := time.Now()

    // Create a new magick
    currentImg, err := magick.New("ghost.jpg", "/tmp")
    if err != nil {
      panic(err)
    }

    // Generate watermark
    result, err := GenerateWatermark(currentImg)
    if err != nil {
      panic(err)
    }

    // Image already persisted to disk
    fmt.Println("Path", result.SourcePath)

    // Calculate elapsed time
    elapsedTime := time.Since(startTime)
    fmt.Println("Elapsed time:", elapsedTime)
  }
}
```

Watermark using ImageMagick

```go
func GenerateWatermark(currentImg *magick.ImageMagick) (*magick.ImageMagick, error) {
  // Add original image
  magickCommands := make([]magick.Command, 0)
  magickCommands = append(magickCommands, magick.AddImage{
    Source:   currentImg.SourcePath,
    Position: magick.Center,
  })

  // Add watermark to the commands
  var opacity float32 = 0.5
  magickCommands = append(magickCommands, magick.AddImage{
    Source:   "watermark.png",
    Position: magick.Center,
    Opacity:  &opacity,
    Resize: &magick.ResizeTarget{
      FixedWidth: int(currentImg.Width() / 2),
    },
  })

  // Generate final image
  result, err := currentImg.Apply(magickCommands...)
  if err != nil {
    return nil, err
  }

  return result, nil
}
```

Image Magick Wrapper

Image Magick

- CLI made for creating, converting, modifying, and editing raster images.
- Released in 1990 (34 years ago)
- Latest stable release 25 February 2024

Image Magick Library

```go
// RunCommands executes the pipeline
// and returns the path to the output file
func (i *ImageMagick) RunCommands(commands ...Command) (string, error) {
  // Mysteries of the universe
}
```

Combine Images

Change quality

Generate Tile

Resize

Image Magick Library

```go
type Command interface {
  Build() []string
}
```

Image Magick Library

Generate Tile Command

```go
type GenerateTile struct {
  Width  int
  Height int
  Source string
}

func (g GenerateTile) Build() []string {
  return []string{
    "-size",
    fmt.Sprintf("%dx%d", g.Width, g.Height),
    fmt.Sprintf("tile:%s", g.Source),
  }
}
```

Image Magick Library

```go
im, err := New(sourcePath, testdata)
assert.NoError(t, err)

// Generate tile background
opacity := float32(0.5)
output, err := im.RunCommands(
  GenerateTile{
    Source: checkeredPath,
    Width:  im.Width() + 50,
    Height: im.Height() + 50,
  },
  AddImage{
    Source:   sourcePath,
    Position: Center,
  },
  AddImage{
    Source:   watermarkPath,
    Position: Center,
    Opacity:  &opacity,
    Resize: &ResizeTarget{
      FixedHeight: im.Height() / 2,
    },
  },
  Resize{
    Target: ResizeTarget{
      FixedHeight: im.Height() / 2,
    },
  },
  Quality{Value: 75},
)
```

Image Magick Library

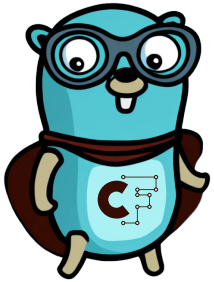
Result of the applying watermark

```
convert (ghost.jpg) -gravity center (watermark.png -resize 2912
-channel A -evaluate multiply 0.5 +channel) -gravity center
-composite output.jpg
```

Image Magick Library

CF Image Magick Wrapper

```
output, err := im.RunCommands(
  GenerateTile{
    Source: checkeredPath,
    Width:  100,
    Height: 100,
  },
)
```



gographics/imagick

```
// make-tile creates a tileable image from an input image.
// ( +clone -flop ) +append  ( +clone -flip ) -append -resize 50%
func make_tile(mw *imagick.MagickWand, outfile string) {
        mwc := mw.Clone()
        mwc.FlopImage()
        mw.AddImage(mwc)
        mwc.Destroy()
        mwc = mw.AppendImages(false)
        mwf := mwc.Clone()
        mwf.FlipImage()
        mwc.AddImage(mwf)
        mwf.Destroy()
        mwf = mwc.AppendImages(true)

        w := mwf.GetImageWidth()
        h := mwf.GetImageHeight()
        // 1 = Don't blur or sharpen image
        mwf.ResizeImage(w/2, h/2, imagick.FILTER_LANCZOS, 1)
        mwf.WriteImage(outfile)
        mwf.Destroy()
        mwc.Destroy()
}
```

Benchmark

Spec:

aws

t3a.medium

2 vCPU // 4GB RAM

Running on separated nodes

```
NAME                                    READY   STATUS    RESTARTS   AGE   IP             NODE
watermark-go-directly-f7fd9fb95-2cq67   1/1     Running   0          11m   10.1.35.126    ip-10-1-47-242.ec2.internal
watermark-go-directly-f7fd9fb95-b49p6   1/1     Running   0          11m   10.1.85.114    ip-10-1-87-231.ec2.internal
watermark-go-directly-f7fd9fb95-cggdm   1/1     Running   0          11m   10.1.109.51    ip-10-1-98-176.ec2.internal
watermark-go-directly-f7fd9fb95-qv7dt   1/1     Running   0          11m   10.1.51.196    ip-10-1-53-178.ec2.internal
watermark-go-directly-f7fd9fb95-rjpmf   1/1     Running   0          11m   10.1.69.197    ip-10-1-75-138.ec2.internal
watermark-go-directly-f7fd9fb95-s5vjr   1/1     Running   0          11m   10.1.73.201    ip-10-1-69-107.ec2.internal
watermark-go-directly-f7fd9fb95-wqb62   1/1     Running   0          11m   10.1.98.91     ip-10-1-107-191.ec2.internal
watermark-go-directly-f7fd9fb95-wqzr7   1/1     Running   0          11m   10.1.82.36     ip-10-1-80-36.ec2.internal
watermark-go-directly-f7fd9fb95-xlv6v   1/1     Running   0          11m   10.1.90.25     ip-10-1-86-149.ec2.internal
watermark-go-directly-f7fd9fb95-xpzms   1/1     Running   0          11m   10.1.71.150    ip-10-1-73-222.ec2.internal
watermark-magick-c6b7458bb-559gw        1/1     Running   0          11m   10.1.111.74    ip-10-1-110-188.ec2.internal
watermark-magick-c6b7458bb-82b2w        1/1     Running   0          11m   10.1.69.84     ip-10-1-77-173.ec2.internal
watermark-magick-c6b7458bb-85mg7        1/1     Running   0          11m   10.1.35.52     ip-10-1-35-161.ec2.internal
watermark-magick-c6b7458bb-8xcxm        1/1     Running   0          11m   10.1.84.144    ip-10-1-85-18.ec2.internal
watermark-magick-c6b7458bb-fg7b8        1/1     Running   0          11m   10.1.50.146    ip-10-1-53-104.ec2.internal
watermark-magick-c6b7458bb-hp4dw        1/1     Running   0          11m   10.1.55.126    ip-10-1-60-233.ec2.internal
watermark-magick-c6b7458bb-hsprq        1/1     Running   0          11m   10.1.100.217   ip-10-1-109-8.ec2.internal
watermark-magick-c6b7458bb-j6glv        1/1     Running   0          11m   10.1.68.108    ip-10-1-69-101.ec2.internal
watermark-magick-c6b7458bb-rbg8w        1/1     Running   0          11m   10.1.104.195   ip-10-1-107-192.ec2.internal
watermark-magick-c6b7458bb-v87xr        1/1     Running   0          11m   10.1.53.224    ip-10-1-51-127.ec2.internal
```

CPU and Memory usage (avg)

```
NAME                                    CPU(cores)   MEMORY(bytes)
watermark-go-directly-f7fd9fb95-2cq67   1403m        1490Mi
watermark-go-directly-f7fd9fb95-b49p6   1323m        1498Mi
watermark-go-directly-f7fd9fb95-cggdm   1285m        1488Mi
watermark-go-directly-f7fd9fb95-qv7dt   1274m        1491Mi
watermark-go-directly-f7fd9fb95-rjpmf   1211m        1494Mi
watermark-go-directly-f7fd9fb95-s5vjr   1353m        1485Mi
watermark-go-directly-f7fd9fb95-wqb62   1275m        1488Mi
watermark-go-directly-f7fd9fb95-wqzr7   1310m        1493Mi
watermark-go-directly-f7fd9fb95-xlv6v   1319m        1493Mi
watermark-go-directly-f7fd9fb95-xpzms   1426m        1493Mi
watermark-magick-c6b7458bb-559gw        867m         1866Mi
watermark-magick-c6b7458bb-82b2w        769m         1869Mi
watermark-magick-c6b7458bb-85mg7        840m         1571Mi
watermark-magick-c6b7458bb-8xcxm        930m         1823Mi
watermark-magick-c6b7458bb-fg7b8        883m         1859Mi
watermark-magick-c6b7458bb-hp4dw        890m         1770Mi
watermark-magick-c6b7458bb-hsprq        982m         1855Mi
watermark-magick-c6b7458bb-j6glv        853m         1563Mi
watermark-magick-c6b7458bb-rbg8w        838m         1493Mi
watermark-magick-c6b7458bb-v87xr        894m         1888Mi
```

Using GO

CPU	~ 1318 m
Memory	~ 1.49 GB

Using Magick

CPU	~ 874 m
Memory	~ 1.75 GB

Processing time

GO

```
Elapsed time: 6.061475131s
Elapsed time: 5.679578247s
Elapsed time: 5.885973701s
Elapsed time: 5.793911165s
Elapsed time: 5.725055295s
Elapsed time: 5.726625929s
Elapsed time: 6.0449784s
Elapsed time: 5.823044497s
Elapsed time: 6.013011328s
Elapsed time: 5.84848784s
Elapsed time: 5.827224173s
Elapsed time: 5.848494815s
Elapsed time: 5.680053485s
Elapsed time: 6.01665353s
Elapsed time: 6.175302275s
Elapsed time: 6.205006762s
Elapsed time: 5.539561962s
Elapsed time: 5.588447152s
Elapsed time: 5.573180269s
```

Image Magick

```
Elapsed time: 26.47917003s
Elapsed time: 29.059188466s
Elapsed time: 31.393784122s
Elapsed time: 29.366977988s
Elapsed time: 30.955103898s
Elapsed time: 30.196296531s
Elapsed time: 29.24341307s
Elapsed time: 28.884622896s
Elapsed time: 29.104793889s
Elapsed time: 30.640078773s
Elapsed time: 28.101539115s
Elapsed time: 28.123198938s
Elapsed time: 28.792203136s
Elapsed time: 29.899890694s
Elapsed time: 29.297364937s
Elapsed time: 30.699912635s
Elapsed time: 29.137336125s
```

Image Magick Wrapper

Veeeery slow

Processing Time

~29s

RAM Required

~1750 MB

CPU Required

~ 874m

Blazingly fast 🏃

Processing Time

~5s

RAM Required

~1490 MB

CPU Required

~ 1300m

Okay, but why?

Are you still using Image Magick?

```
  0[||||||||||||||||||||||||||||||||||||                                    43.0%] Tasks: 3, 4 thr, 0 kthr; 2 running
  1[||||||||||||||||||||||||||||||||||||||||||||||||||||||                   62.0%] Load average: 2.34 1.89 1.46
Mem[||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||1.79G/3.77G] Uptime: 00:43:56
Swp[                                                                          0K/0K]

 Main   I/O
  PID USER      PRI  NI  VIRT   RES   SHR S  CPU%▽MEM%   TIME+  Command
    1 root       20   0 2863M 1480M  1588 S  94.4 38.3  5:19.29 /watermark
   10 root       20   0 2863M 1480M  1588 R  56.8 38.3  2:06.40 /watermark
    8 root       20   0 2863M 1480M  1588 S  35.6 38.3  1:50.21 /watermark
    7 root       20   0 2863M 1480M  1588 S   0.7 38.3  0:00.98 /watermark
    9 root       20   0 2863M 1480M  1588 S   0.0 38.3  0:00.00 /watermark
   11 root       20   0  1672  1008   868 S   0.0  0.0  0:00.01 sh
   22 root       20   0  2096  1828  1012 R   0.0  0.0  0:00.08 htop
```

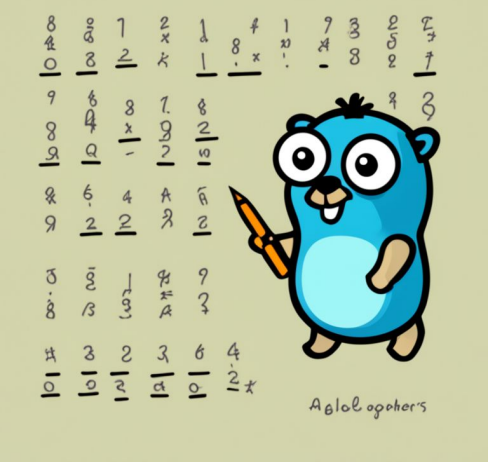
GO Directly

```
  0[||||||||||||||||                                                        18.8%] Tasks: 4, 4 thr, 0 kthr; 2 running
  1[||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| 89.4%] Load average: 1.00 1.28 1.29
Mem[||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||1.93G/3.77G] Uptime: 00:30:10
Swp[                                                                          0K/0K]

 Main   I/O
  PID USER      PRI  NI  VIRT   RES   SHR S  CPU%▽MEM%   TIME+  Command
   48 root       20   0 1970M 1632M  4932 R  97.7 42.3  0:09.19 convert ( ghost.jpg ) -gravity center ( watermark.png -resize 2912 -channel A -evaluate multiply 0.5 +cha
    1 root       20   0 1197M  3056  1868 S   0.0  0.1  0:00.05 /watermark
    7 root       20   0 1197M  3056  1868 S   0.0  0.1  0:00.02 /watermark
    8 root       20   0 1197M  3056  1868 S   0.0  0.1  0:00.00 /watermark
    9 root       20   0 1197M  3056  1868 S   0.0  0.1  0:00.00 /watermark
   10 root       20   0 1197M  3056  1868 S   0.0  0.1  0:00.00 /watermark
   28 root       20   0  1672  1140   996 S   0.0  0.0  0:00.01 sh
   39 root       20   0  1852  1620  1048 R   0.0  0.0  0:00.07 htop
```

Image Magick

```
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```



> 29.1 MM

Assets

> 4.1 MM

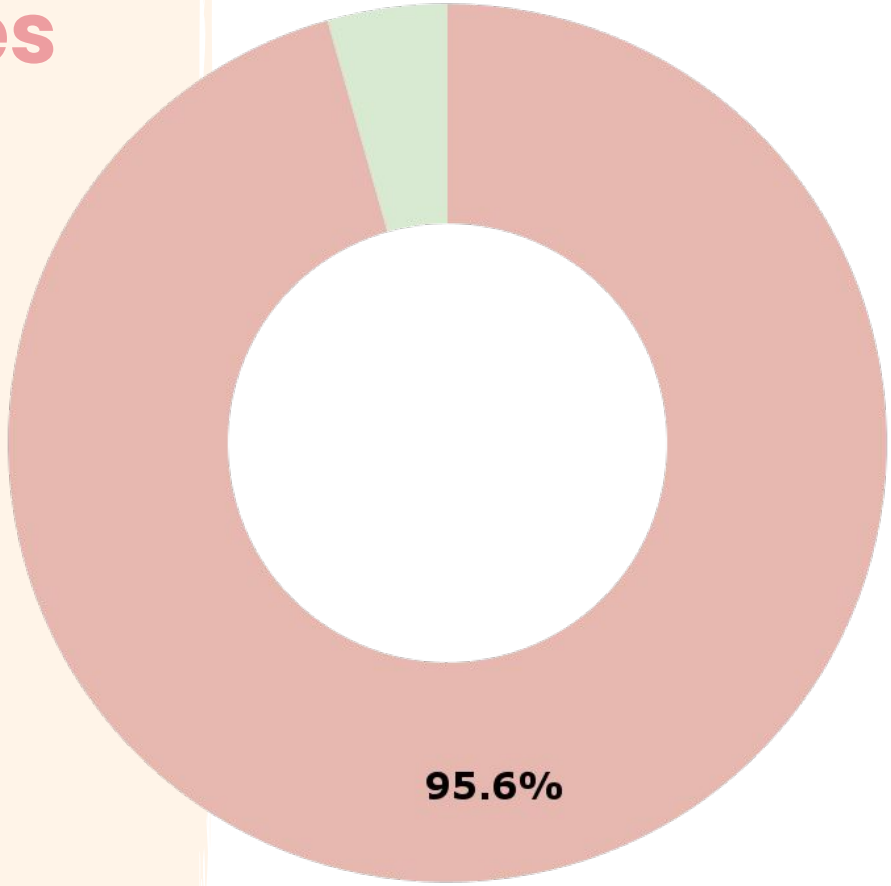
Products

> 9.4 MM

Curations

Categories

Most of the current content is coming from the **PNG** (illustrations) category.



PNG
Main category for now.

Others
Stockphotos, patterns, backgrounds, mockups...

PNGs require mostly generating tiles

Checkered Background

Used when generating previews for PNGs illustrations.

Creative Fabrica is a Startup

We gotta move the product fast!

- Fast iterations
- Magick support 200+ file formats
- Developer doesn't need to know pixel-space manipulation
- Go officially doesn't support many image types (yet)
- Reuse same commands when dealing with different image formats

Tracing

Trace

cf_products: cf_products.consumer.preview_generator 2.89s

2024-02-26 13:55:07.766

Give feedback | Trace ID | Export

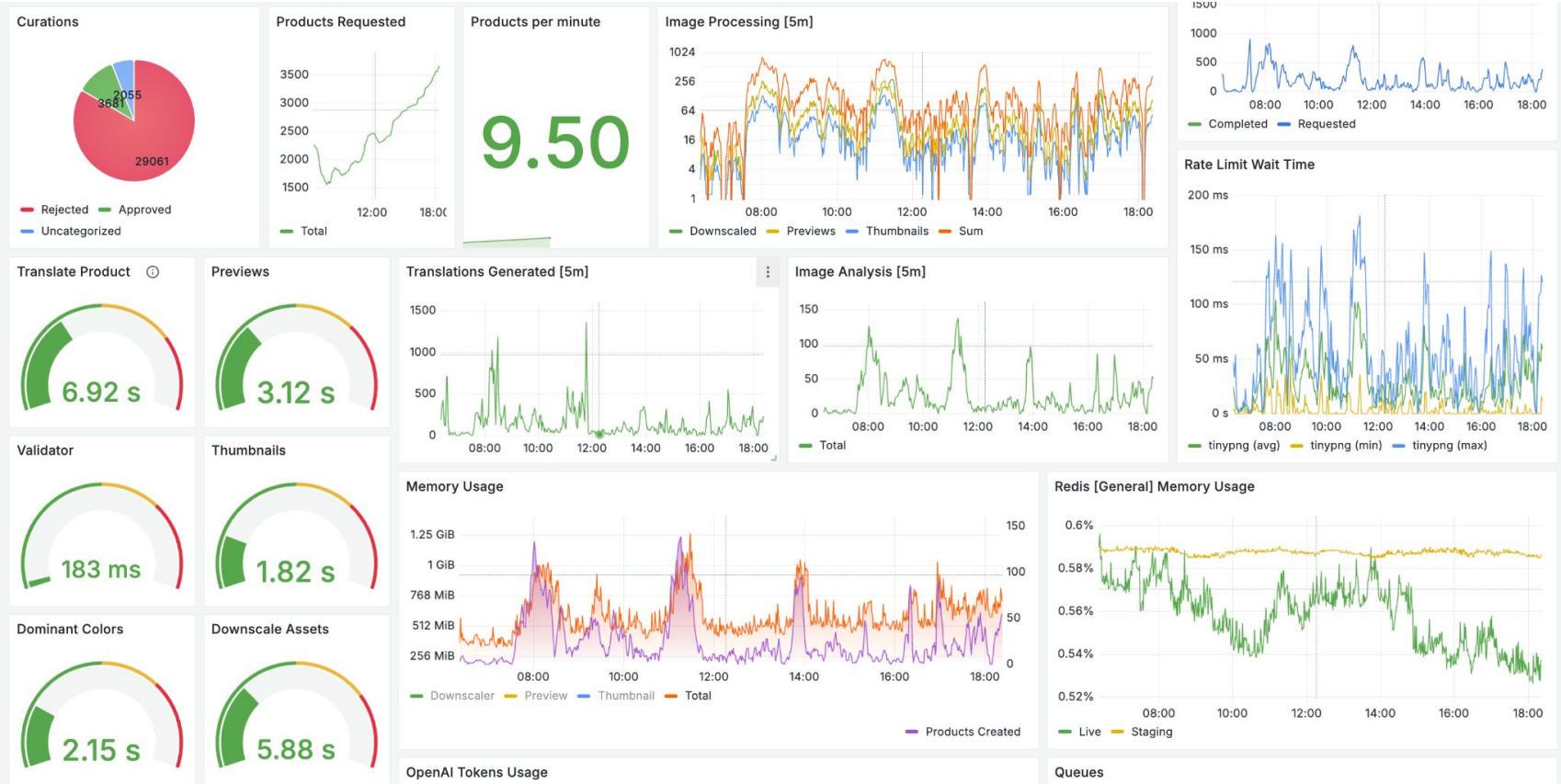
Span Filters

16 spans | Prev | Next

0µs | 722.41ms | 1.44s | 2.17s | 2.89s

Service & Operation	0µs	722.41ms	1.44s	2.17s	2.89s
cf_products cf_products.consumer.preview_generator (2.89s)					
cf_products.s3.downloader (18.65ms)	18.65ms				
cf_products.s3.uploader (1.17s)			1.17s		
cf_products.ratelimit.acquire_blocking.tinypng (659.68µs)			659.68µs		
cf_products.s3.uploader (1.35s)			1.35s		
cf_products.ratelimit.acquire_blocking.tinypng (680.51µs)			680.51µs		
cf_products.handler.CreateAssetCommand (16.3ms)				16.3ms	
cf_products.repository.asset.create (11.16ms)				11.16ms	
cf_products.cache.invalidate_item (1.04ms)				1.04ms	
cf_products.repository.counter.create (5.07ms)				5.07ms	
cf_products.producer.command_handler (13.44ms)				13.44ms	
cf_products.handler.CreateAssetCommand (10.26ms)					10.26ms
cf_products.repository.asset.create (5.39ms)					5.39ms
cf_products.cache.invalidate_item (843.27µs)					843.27µs
cf_products.repository.counter.create (4.84ms)					4.84ms
cf_products.producer.command_handler (13.67ms)					13.67ms

Metrics Dashboard



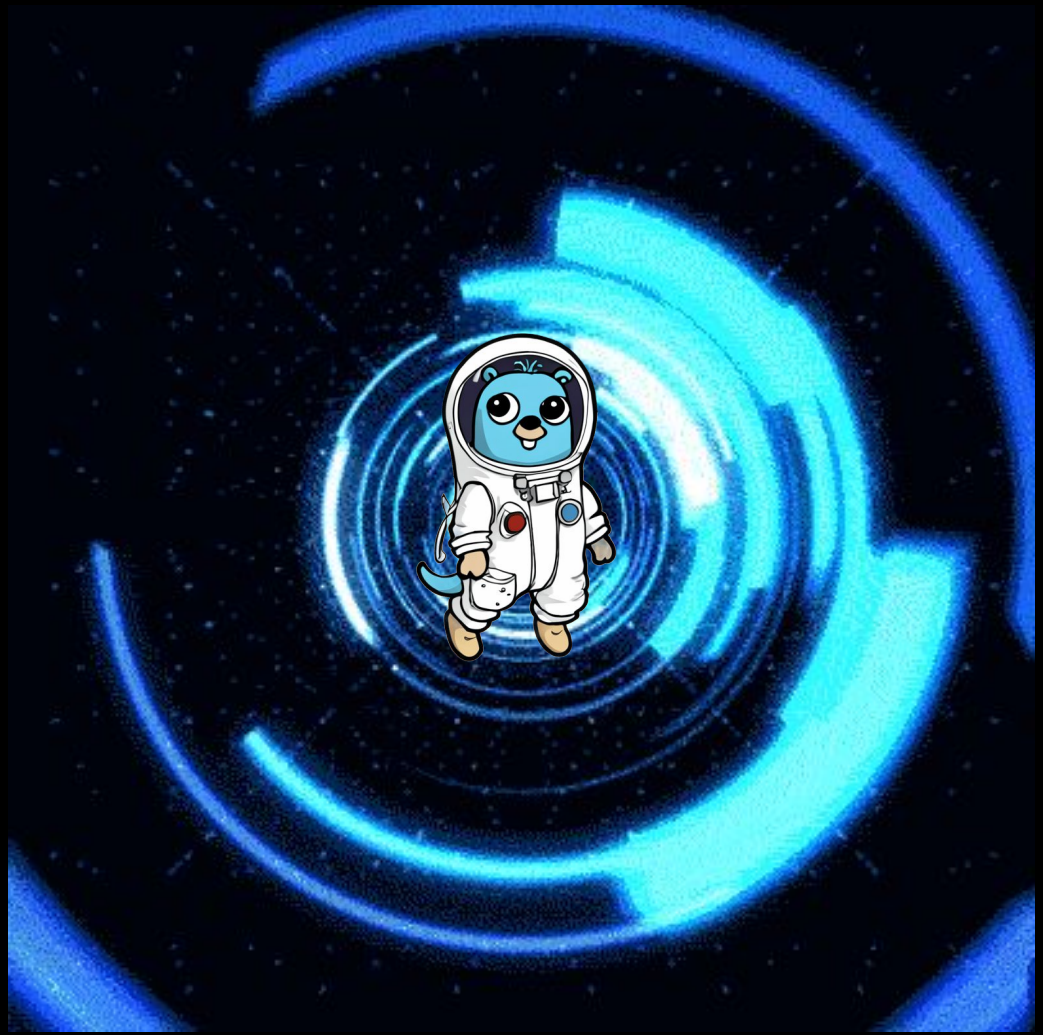
WE ARE HIRING

Are **you** our next gopher?

creativefabrica.com/jobs

Questions?

BONUS TIME!!



Creative Fabrica

GOPHER Verse

go blue gopher wearing futuristic glasses with serious face with a cyberpunk futuristic landscape behind

Share with friends

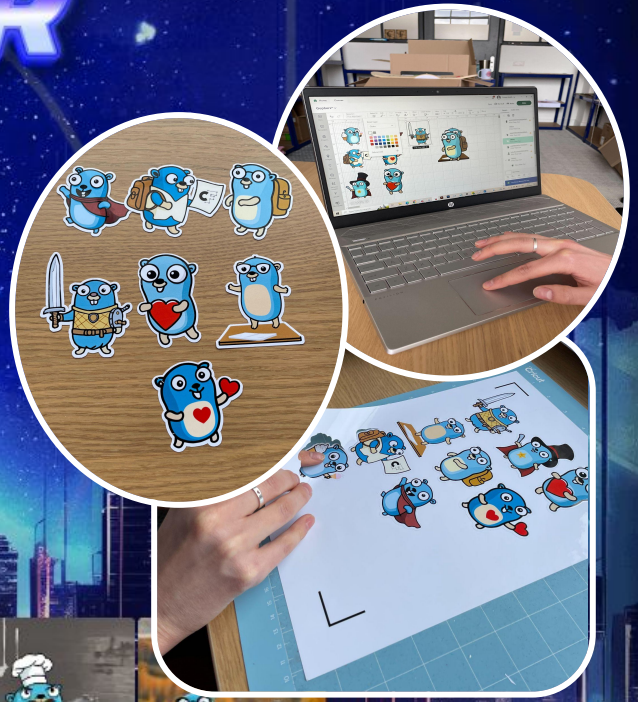
Creative Fabrica

GOPHER Verse

tech.creativefabrica.com

What kind of gopher do you want to generate?

happy gopher in the forest

Generate

Latest Generations