



RUBY ON RAILS Level UP

6. Krok dalej



Praca domowa

RUBY ON RAILS
Level UP ↗

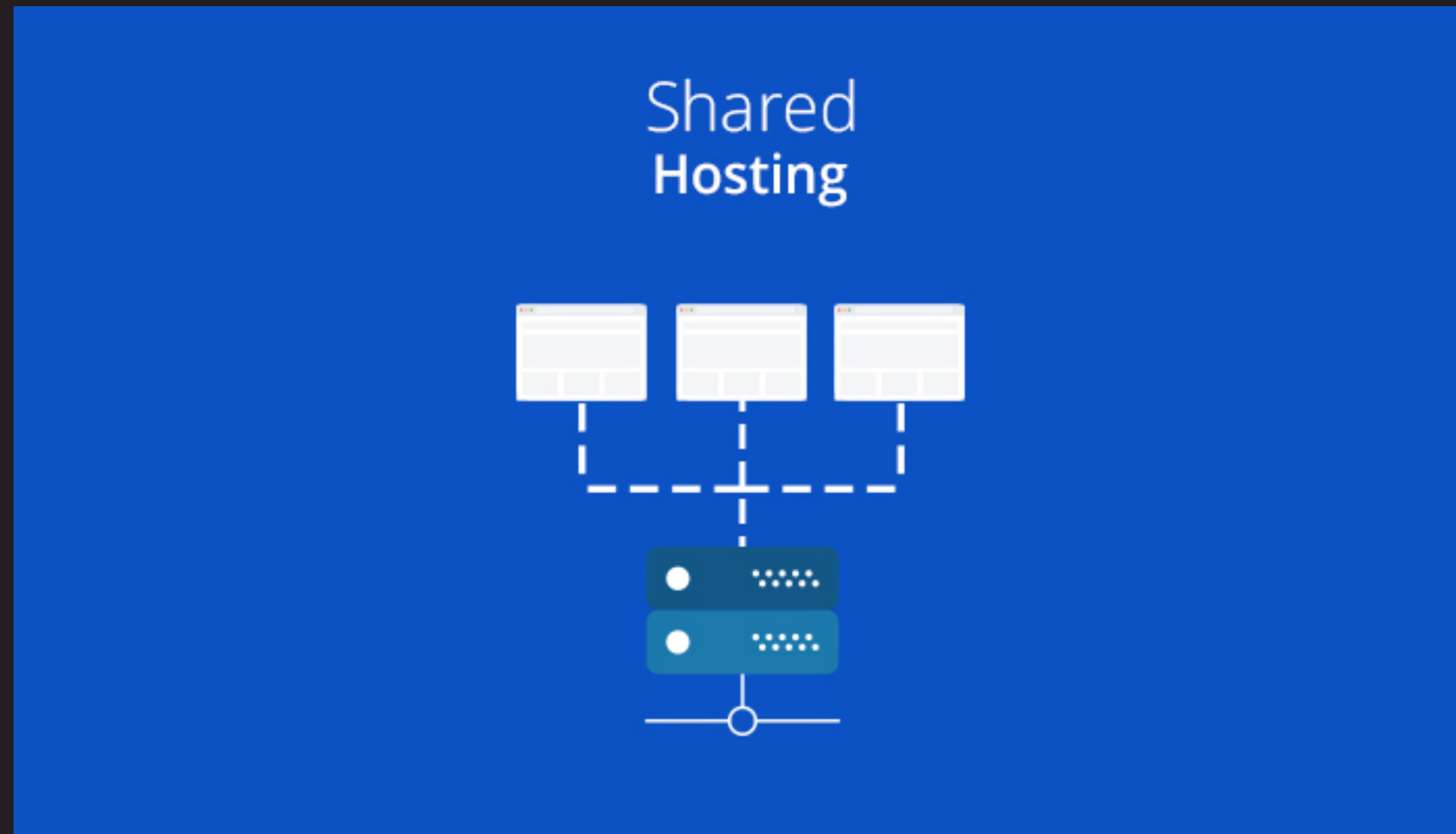
Deploy aplikacji

RUBY ON RAILS
Level UP ↗

Gdzie żyją aplikacje?

- Shared hosting
- Dedicated server
- VPS
- Cloud
- PaaS

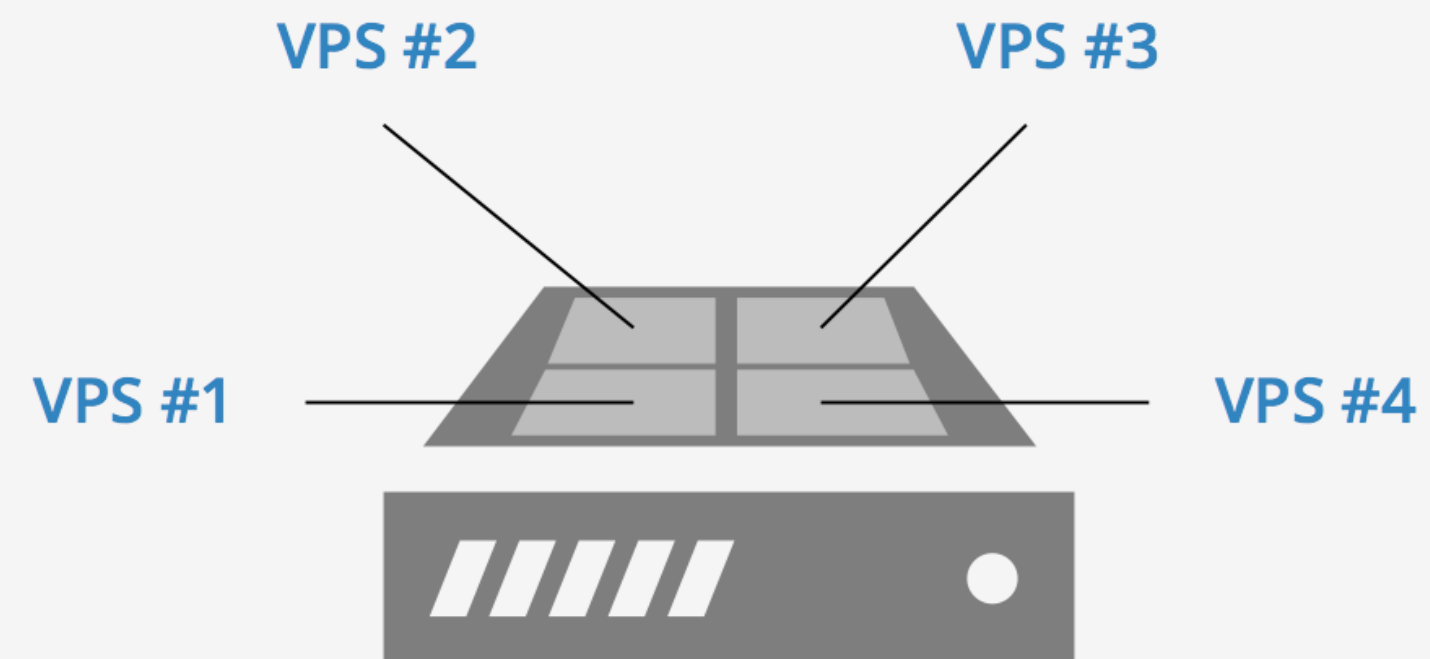
Shared hosting



Dedicated server



VPS



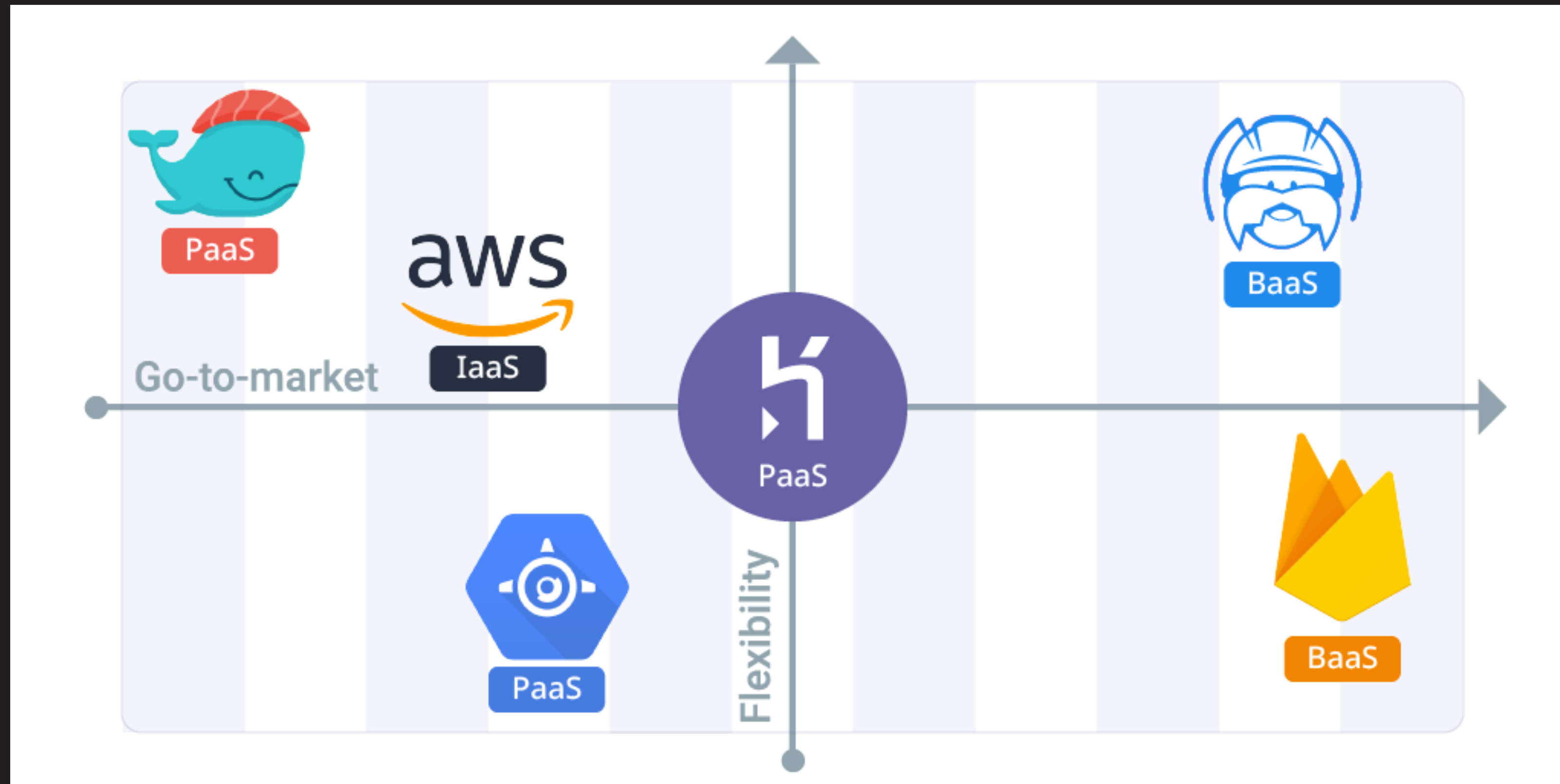
What is a Virtual Private Server?

Cloud

Cloud
Hosting



PaaS - Platform as a Service



Heroku

Create New App


App name

rails-levelup-2019


✓

rails-levelup-2019 is available

Choose a region

 Europe

⌵

 Add to pipeline...

Create app

Heroku

Deployment method



Heroku Git



GitHub



Container Registry

Heroku

Deploy using Heroku Git

Use git in the command line or a GUI tool to deploy this app.

Install the Heroku CLI

Download and install the [Heroku CLI](#).

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

Heroku

Deploy using Heroku Git

Use git in the command li

Install the Heroku CLI

Download and install the

If you haven't already, log

```
$ heroku login
```



HEROKU

Logged in

You can close this page and return to your CLI. It should now be logged in.

e a new SSH public key.

Heroku

Existing Git repository

For existing repositories, simply add the `heroku` remote

```
$ heroku git:remote -a rails-levelup-2019
```

Deploy your application

Commit your code to the repository and deploy it to Heroku using Git.

```
$ git add .  
$ git commit -am "make it better"  
$ git push heroku master
```


Heroku

```
► git push heroku master
Enumerating objects: 111, done.
Counting objects: 100% (111/111), done.
Delta compression using up to 4 threads
Compressing objects: 100% (96/96), done.
Writing objects: 100% (111/111), 1.13 MiB | 832.00 KiB/s, done.
Total 111 (delta 2), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Ruby app detected
remote: -----> Compiling Ruby/Rails
...
remote: -----> Launching...
remote:      Released v6
remote:      https://rails-levelup-2019.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
```

Heroku

► heroku logs

```
2019-04-17T10:18:55.137885+00:00 app[web.1]: F, [2019-04-17T10:18:55.137835 #4]
FATAL -- : [f597394e-b2de-474f-a579-847c845a5e4e] ActiveRecord::StatementInvalid
(PG::UndefinedTable: ERROR:  relation "clans" does not exist
2019-04-17T10:18:55.137887+00:00 app[web.1]: LINE 1: SELECT "clans".* FROM "clans"
2019-04-17T10:18:55.137888+00:00 app[web.1]: ^
2019-04-17T10:18:55.137889+00:00 app[web.1]: : SELECT "clans".* FROM "clans")):
```


Heroku

► heroku logs

```
2019-04-17T10:18:55.137885+00:00 app[web.1]: F, [2019-04-17T10:18:55.137835 #4]
FATAL -- : [f597394e-b2de-474f-a579-847c845a5e4e] ActiveRecord::StatementInvalid
(PG::UndefinedTable: ERROR:  relation "clans" does not exist
2019-04-17T10:18:55.137887+00:00 app[web.1]: LINE 1: SELECT "clans".* FROM "clans"
2019-04-17T10:18:55.137888+00:00 app[web.1]: ^
2019-04-17T10:18:55.137889+00:00 app[web.1]: : SELECT "clans".* FROM "clans"):
```

► heroku run rake db:migrate

```
...
Running rake db:migrate on  rails-levelup-2019... up, run.4573 (Free)
...
```

Heroku

```
▶ curl -s https://rails-levelup-2019.herokuapp.com/clans | jq -c
```

```
{  
  "data": [  
    {  
      "id": "1",  
      "type": "clans",  
      "attributes": {  
        "name": "The very first clan"  
      },  
      "relationships": {  
        "warriors": {  
          "data": [  
            {  
              "id": "1",  
              "type": "warriors-samurais"  
            },  
            ...  
          ]  
        }  
      }  
    }  
  ]  
}
```

Zarządzanie sekretami

Gdzie schować sekrety?

- Zmienne środowiskowe
- `credentials.yml.enc`
- `config/gem`

Zmienne środowiskowe

► heroku config

=== rails-levelup-2019 Config Vars

DATABASE_URL: postgres://...

LANG: en_US.UTF-8

RACK_ENV: production

RAILS_ENV: production

RAILS_LOG_TO_STDOUT: enabled

RAILS_SERVE_STATIC_FILES: enabled


SECRET_KEY_BASE: ...

► heroku config:set SENTRY_DSN=https://...

Setting SENTRY_DSN and restarting  rails-levelup-2019... done, v8

SENTRY_DSN: https://...

Zmienne środowiskowe

```
► heroku run rails console  
Running rails console on  rails-levelup-2019... up, run.4476 (Free)  
Loading production environment (Rails 5.2.2.1)  
irb(main):001:0> ENV.fetch('SENTRY_DSN')  
=> "https://..."
```

credentials.yml.enc

```
▶ rails credentials:edit
```

```
...
```

```
sentry:
```

```
  dsn: https://...
```

```
> Rails.application.credentials.sentry[:dsn]
```

```
=> "https://..."
```

credentials.yml.enc

```
► rails credentials:edit  
...  
production:  
  sentry:  
    dsn: https://...  
staging:  
  sentry:  
    dsn: https://...
```


config gem

```
gem 'config'
```

```
► rails g config:install  
  create  config/initializers/config.rb  
  create  config/settings.yml  
  create  config/settings.local.yml  
  create  config/settings  
  create  config/settings/development.yml  
  create  config/settings/production.yml  
  create  config/settings/test.yml  
  append  .gitignore
```

config gem

```
config/settings.yml  
config/settings/#{environment}.yml  
config/environments/#{environment}.yml
```

```
config/settings.local.yml  
config/settings/#{environment}.local.yml  
config/environments/#{environment}.local.yml
```

Integracja z API

RUBY ON RAILS
Level UP ↗

Połączenie z Internetem

- Net::HTTP
- httparty

Nasze pierwsze zapytanie

```
response = HTTParty.get('https://rails-levelup-2019.herokuapp.com/clans')
puts response.code, response.body
# 200
# {"data":[{"id":"1","type":"clans","attributes":{"name":"The very first clan"},"relationships":
{"warriors":{"data":[{"id":"1","type":"warriors-samurais"}, {"id":"2","type":"warriors-samurais"},
{"id":"4","type":"warriors-hussars"}]}}}, {"id":"2","type":"clans","attributes":{"name":"Training
clan"},"relationships":{"warriors":{"data":[{"id":"3","type":"warriors-samurais"},
{"id":"5","type":"warriors-hussars"}]}}}]}
```

Nasze pierwsze zapytanie

```
response = HTTParty.get('https://rails-levelup-2019.herokuapp.com/clans')
pp JSON.parse(response.body)
# {"data"=>
#   [{"id"=>"1",
#     "type"=>"clans",
#     "attributes"=>{"name"=>"The very first clan"},
#     "relationships"=>
#       {"warriors"=>
#         {"data"=>
#           [{"id"=>"1", "type"=>"warriors-samurais"},
#            {"id"=>"2", "type"=>"warriors-samurais"},
#            {"id"=>"4", "type"=>"warriors-hussars"}]}}}
#   ...
pp response.parsed_response
# {"data"=>
#   ...
```

Nasze pierwsze zapytanie, ale lepiej

```
class ShogunApi
  include HTTParty
  base_uri 'rails-levelup-2019.herokuapp.com'

  def clans
    self.class.get('/clans').parsed_response
  end
end

api = ShogunApi.new
pp api.clans
```

Nasze pierwsze zapytanie, ale lepiej

```
class ShogunApi
  include HTTParty
  base_uri 'rails-levelup-2019.herokuapp.com'
  parser ->(body, _format) { JSON.parse(body)['data'] }

  def clans
    self.class.get('/clans').parsed_response
  end
end
```

```
api = ShogunApi.new
pp api.clans.first
# {"id"=>"1",
#  "type"=>"clans",
#  "attributes"=>{"name"=>"The very first clan"},
#  "relationships"=>
#    {"warriors"=>
#      {"data"=>
#        ...
```


Nasze pierwsze zapytanie, ale lepiej

```
class ShogunApi
  include HTTParty
  base_uri 'rails-levelup-2019.herokuapp.com'
  parser ->(body, _format) { JSON.parse(body)['data'] }

  def clans
    self.class.get('/clans').parsed_response
  end

  def clan(id)
    self.class.get("/clans/#{id}").parsed_response
  end

  def create_clan(name:)
    self.class.post('/clans', body: { name: name }).parsed_response
  end
end
```

Nasze pierwsze zapytanie, ale lepiej

```
api = ShogunApi.new
pp api.create_clan(name: 'very good clan, please join')
# {"id"=>"3",
#  "type"=>"clans",
#  "attributes"=>{"name"=>"very good clan, please join"},
#  "relationships"=>{"warriors"=>{"data"=>[]}}}
```

Obsługa błędów API

```
api = ShogunApi.new
```

```
pp api.clan('whatever')  
# nil
```

```
pp api.create_clan(name: nil)  
# nil
```

Obsługa błędów API

```
class ShogunApi
  ...
  def create_clan(name:)
    handle_response(self.class.post('/clans', body: { name: name }))
  end

  private

  def handle_response(response)
    case response.code
    when 200..299
      response.parsed_response
    when 404
      raise RecordNotFound, response.body
    when 422
      raise RecordInvalid, response.body
    else
      raise SomethingIsReallyWrong, response.body
    end
  end
end
```

Obsługa błędów API

```
api = ShogunApi.new
```

```
pp api.clan('whatever')
```

```
# RecordNotFound ({"error": "Couldn't find Clan with 'id'=whatever"})
```

```
pp api.create_clan(name: nil)
```

```
# RecordInvalid ({"name": ["can't be blank"]})
```

Obsługa błędów API

```
module HTTParty
  class Error < StandardError; end
  class UnsupportedFormat < Error; end
  class UnsupportedURIScheme < Error; end
  class ResponseError < Error; attr_reader :response; end
  class RedirectionTooDeep < ResponseError; end
  class DuplicateLocationHeader < ResponseError; end
end
```

```
module Net
  class ProtocolError < StandardError; end
  class ProtoFatalError < ProtocolError; end
  class HTTPError < ProtocolError; end
  class HTTPFatalError < ProtoFatalError; end
  ...
end
```

...

Jak to testować?

- WebMock
- VCR

WebMock

```
require 'webmock/rspec'
```

```
WebMock.disable_net_connect!
```

```
stub_request(:post, 'rails-levelup-2019.herokuapp.com/clans')  
  .with(body: { name: 'name from test' })  
  .to_return(body: { id: 1, type: 'clans', ... })
```


Jak to testować?

```
require 'vcr'

VCR.configure do |config|
  config.cassette_library_dir = 'fixtures/vcr_cassettes'
  config.hook_into :webmock
end

VCR.use_cassette('shogunate') do
  response = HTTParty.get('https://rails-levelup-2019.herokuapp.com/clans')
  expect(response.body)...
end
```



Thanks!