KROMESCH SÁNDOR

APP FELHŐ

API-k és Webszolgáltatások a Cloudban

Magyarországi Web Konferencia 2014 - November 8.

ERISE

Kiindulás: Mobil eszközök 1/2

Mobil eszközök

1. Hordozhatók
2. Könnyen kezelhetők
3. Limitált erőforrásokkal rendelkeznek
4. Hozzáférnek az Internethez

ERiSE

Kiindulás: Cloud 2/2

Szerverek ezrei

1. “korlátlan” erőforrások
2. hozzáférnek az Internethez

Kiindulás: Cloud 2/2

Szerverek ezrei

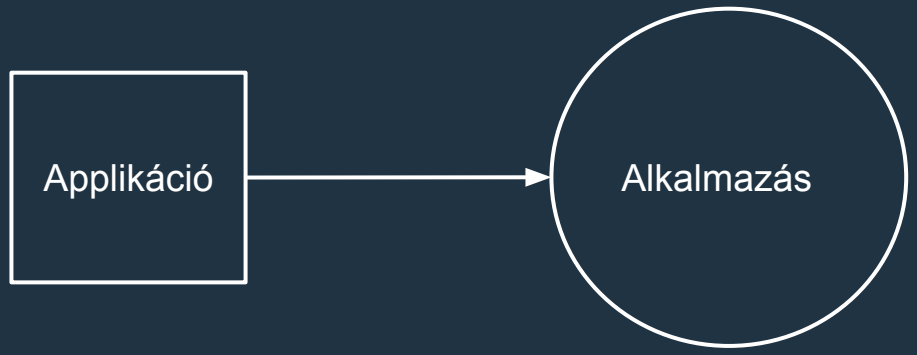
1. “korlátlan” erőforrások
2. hozzáférnek az Internethez

Szerverek száma:

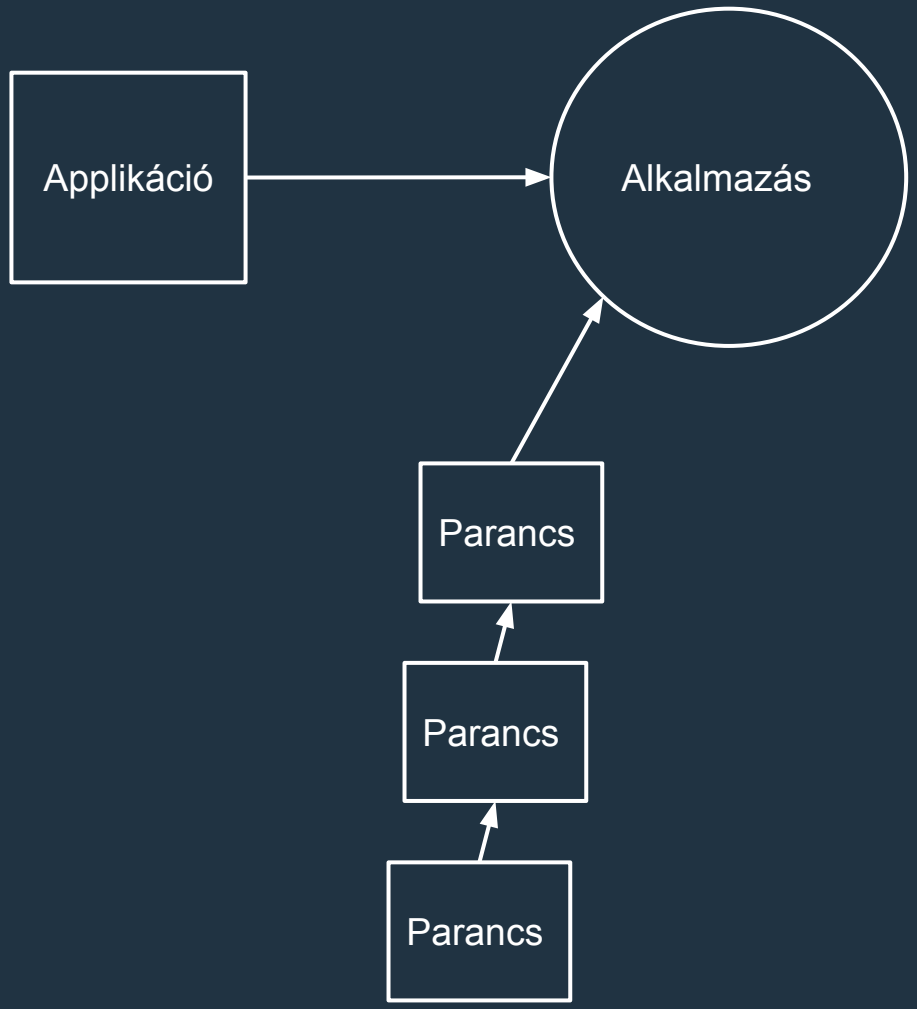
Amazon: 150e - 400e

Google: 1.500e+

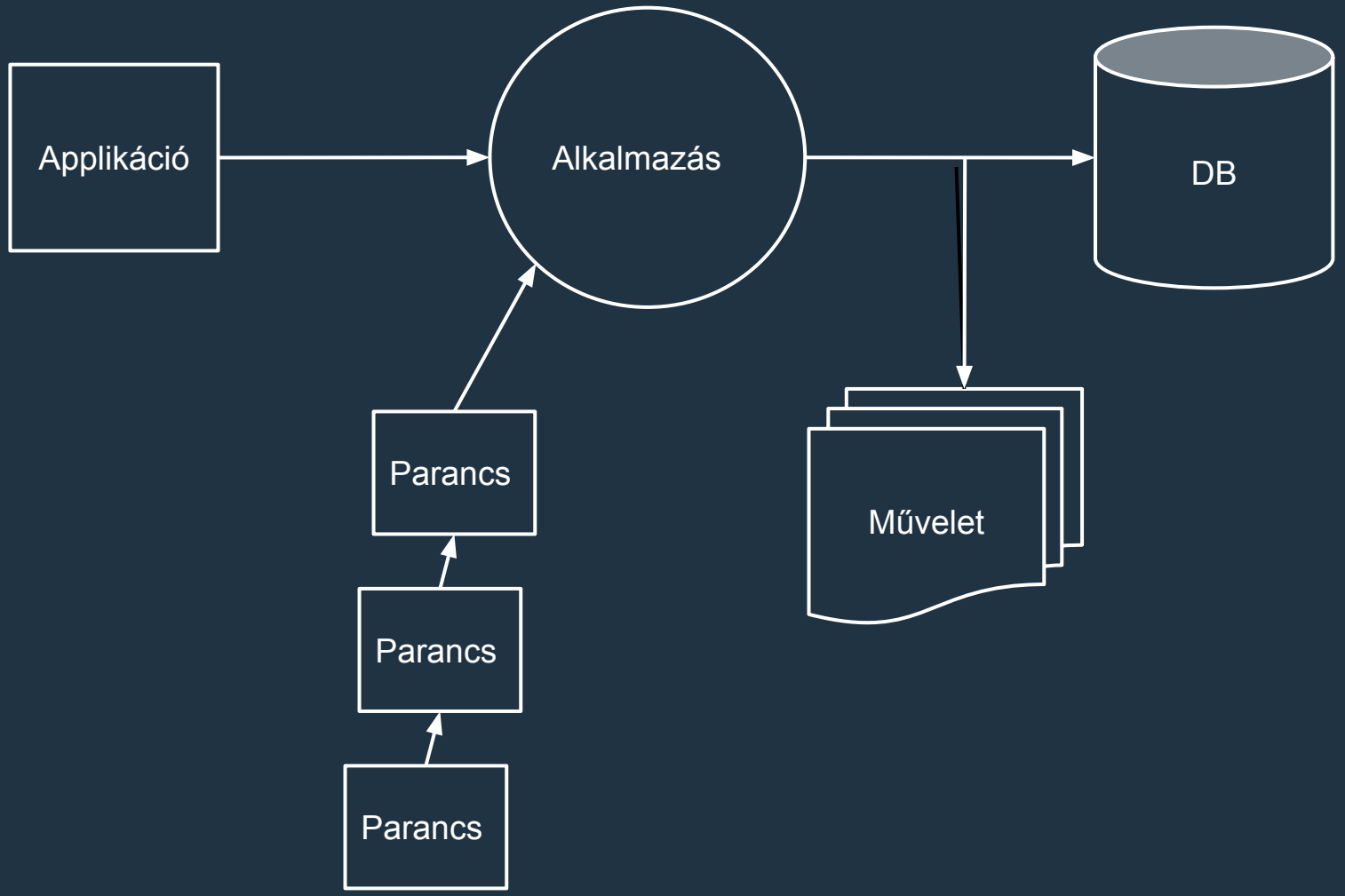
Mi a Cloud célja?



Mi a Cloud célja?

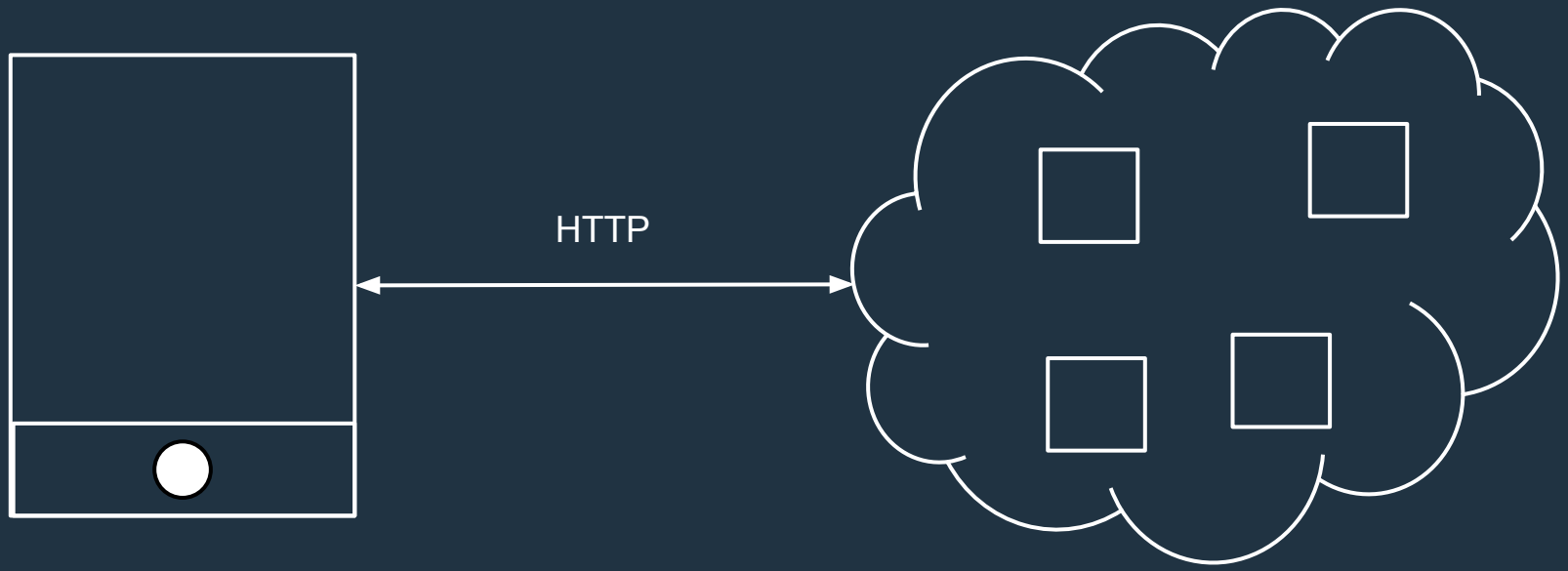


Mi a Cloud célja?



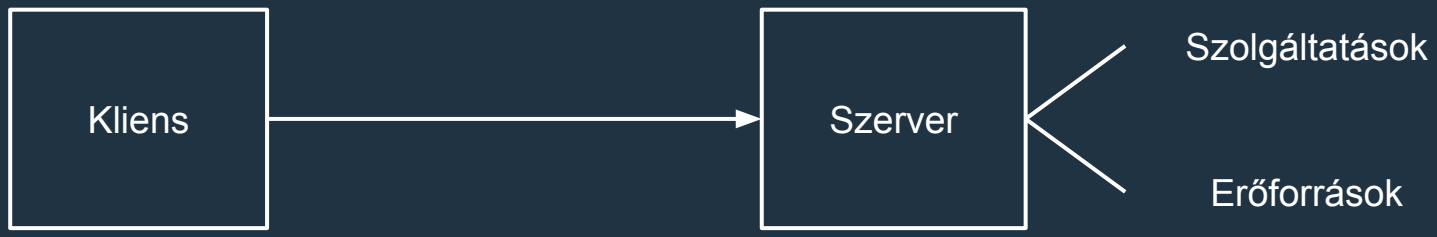
Miért HTTP?

Miért használjunk web böngészőkhöz kifejlesztett protocolt a Mobil eszközökkel és a Cloud szolgáltatásokkal?

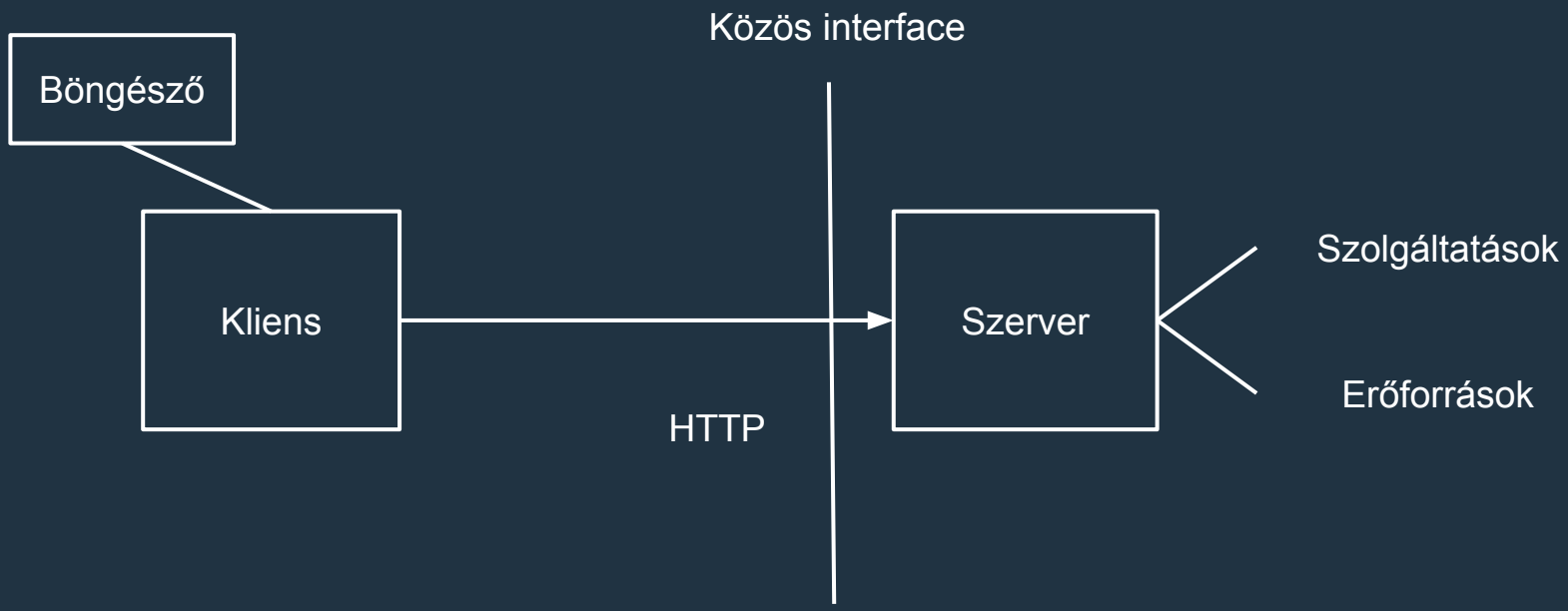


Miért HTTP?

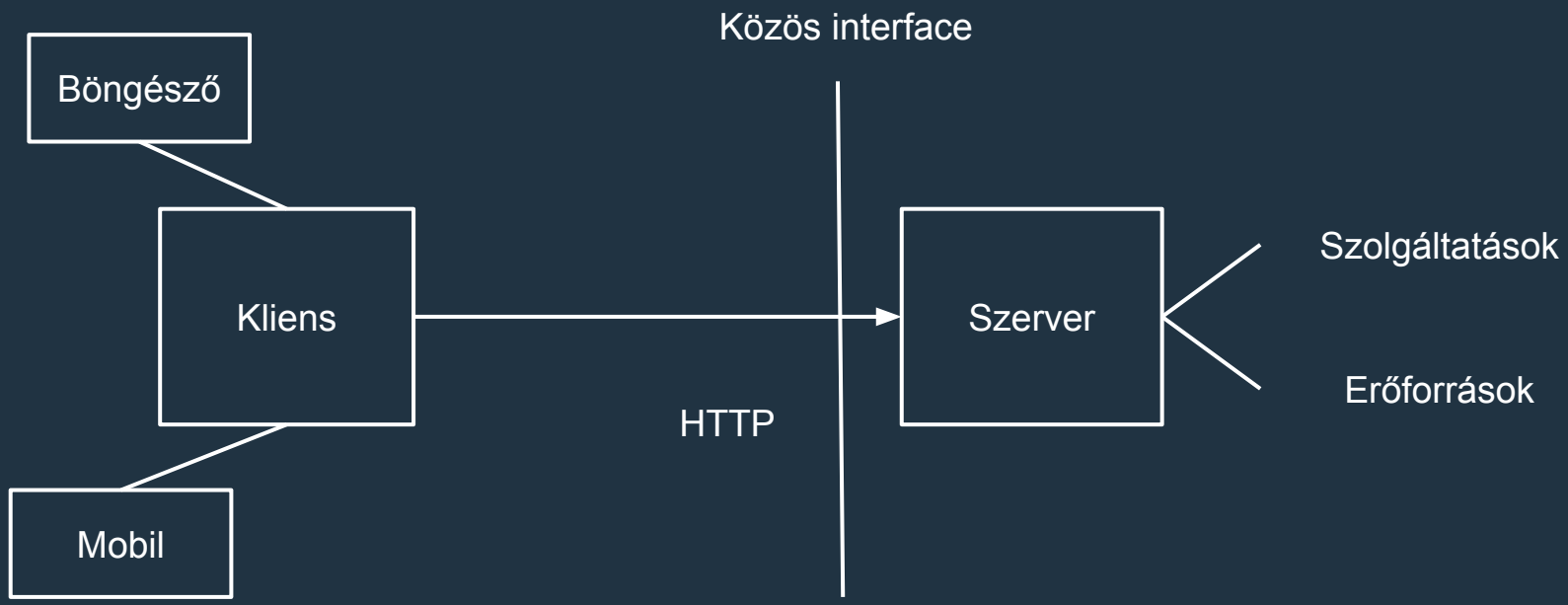
Miért HTTP?



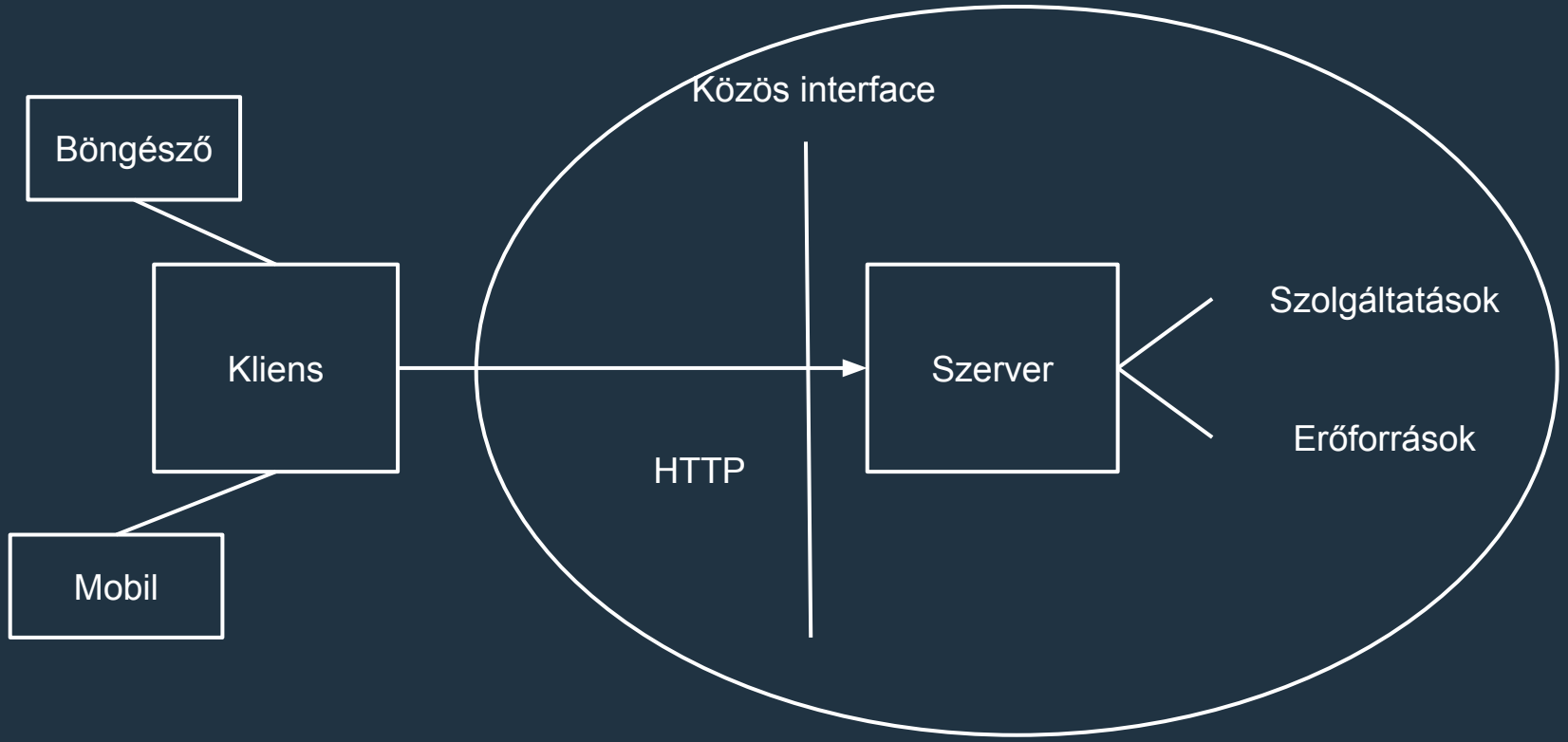
Miért HTTP?


Közös interface
Böngésző
Kliens
Szerver
Szolgáltatások
Erőforrások
HTTP

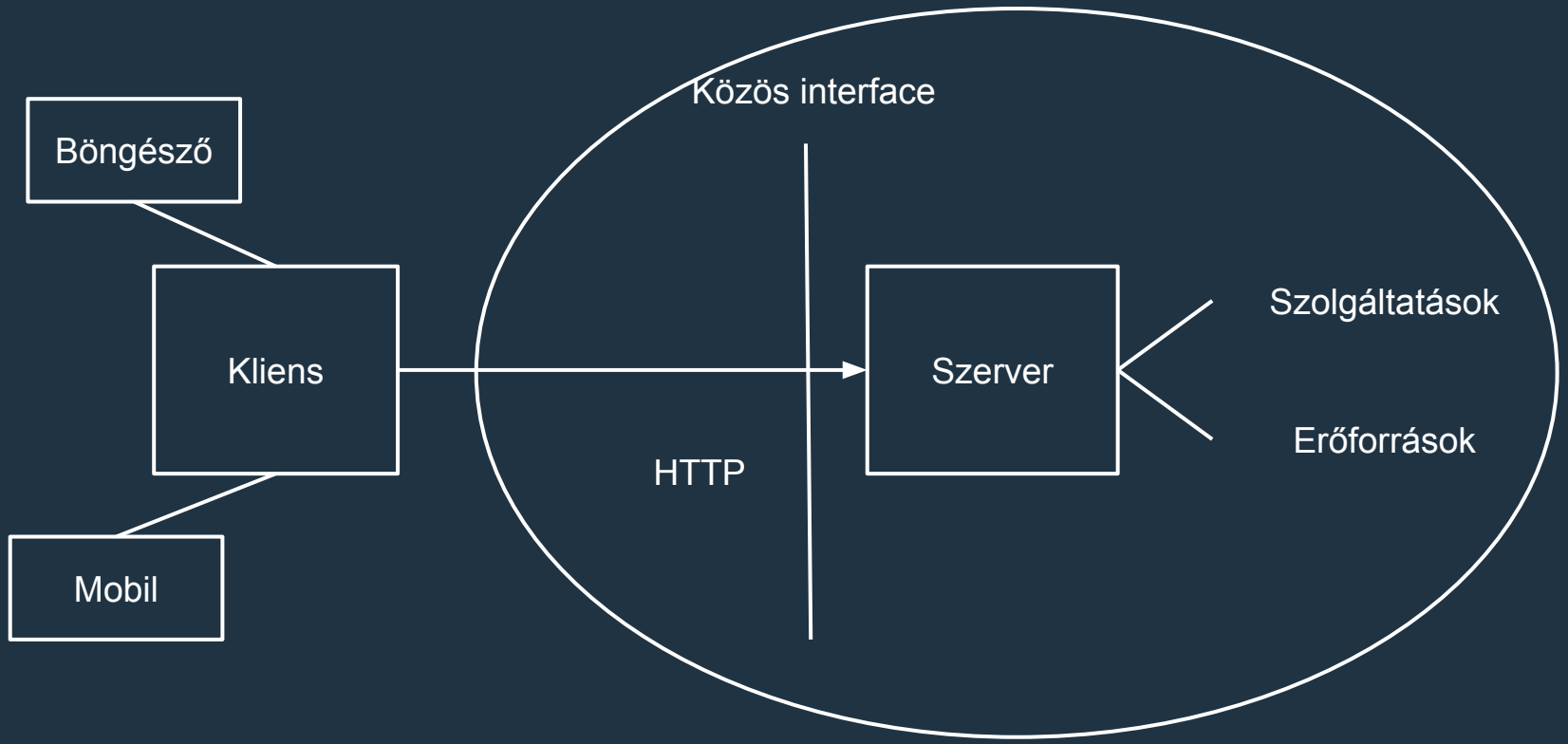
Miért HTTP?


Közös interface
Böngésző
Kliens
Mobil
HTTP
Szerver
Szolgáltatások
Erőforrások

Miért HTTP?

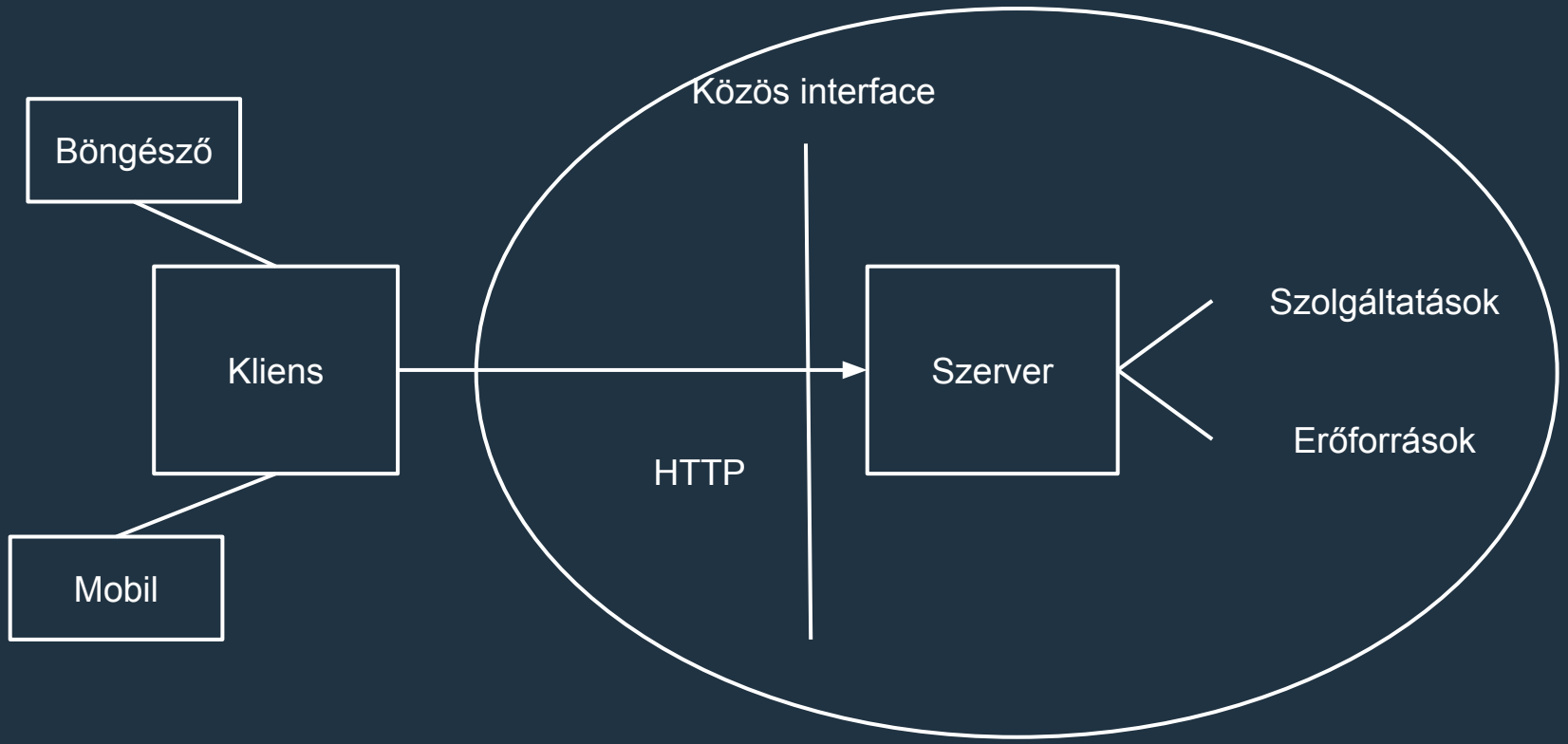

Közös interface
Böngésző
Kliens
Mobil
HTTP
Szerver
Szolgáltatások
Erőforrások

Miért HTTP?



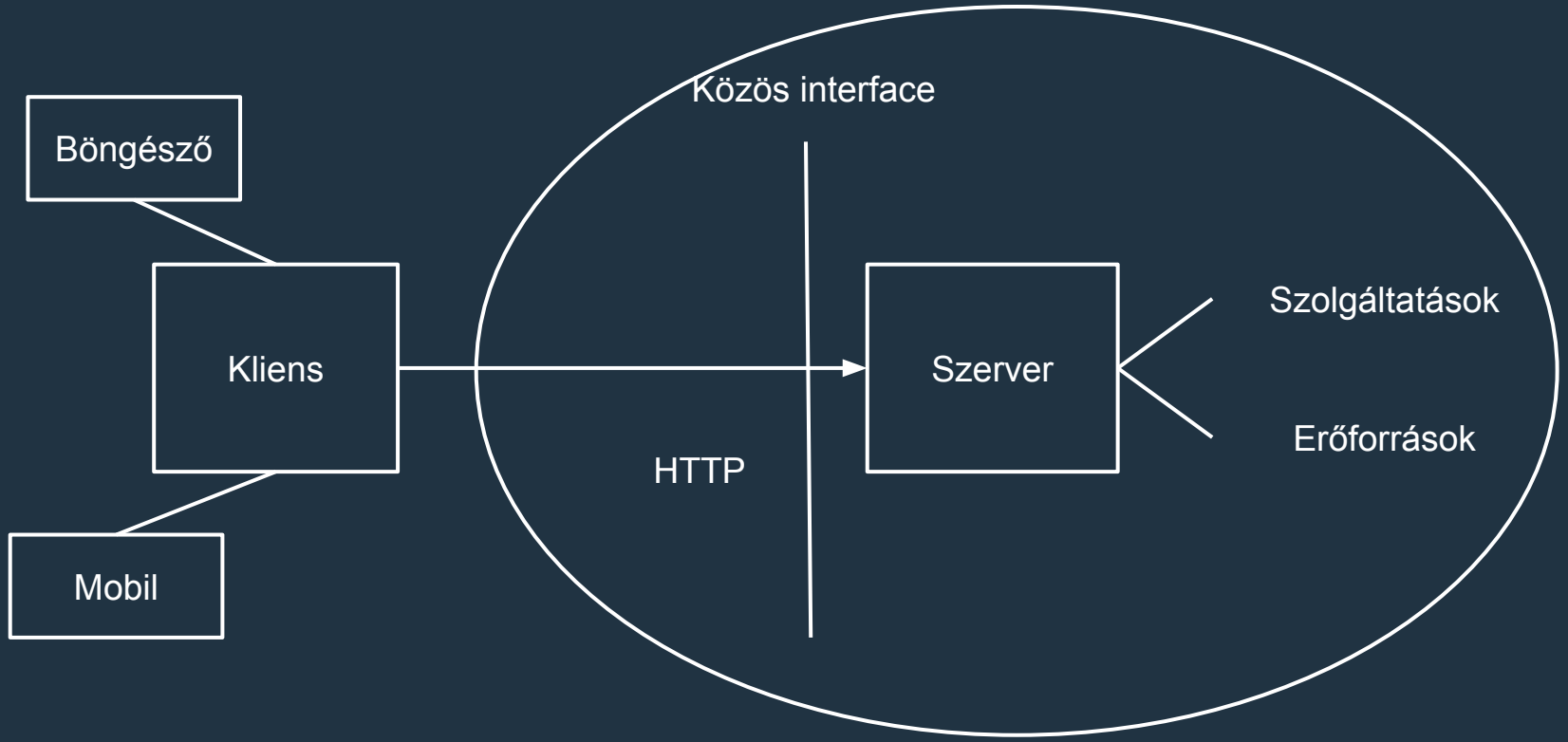
- Frameworkok

Miért HTTP?



- Frameworkok
- Data marshaling

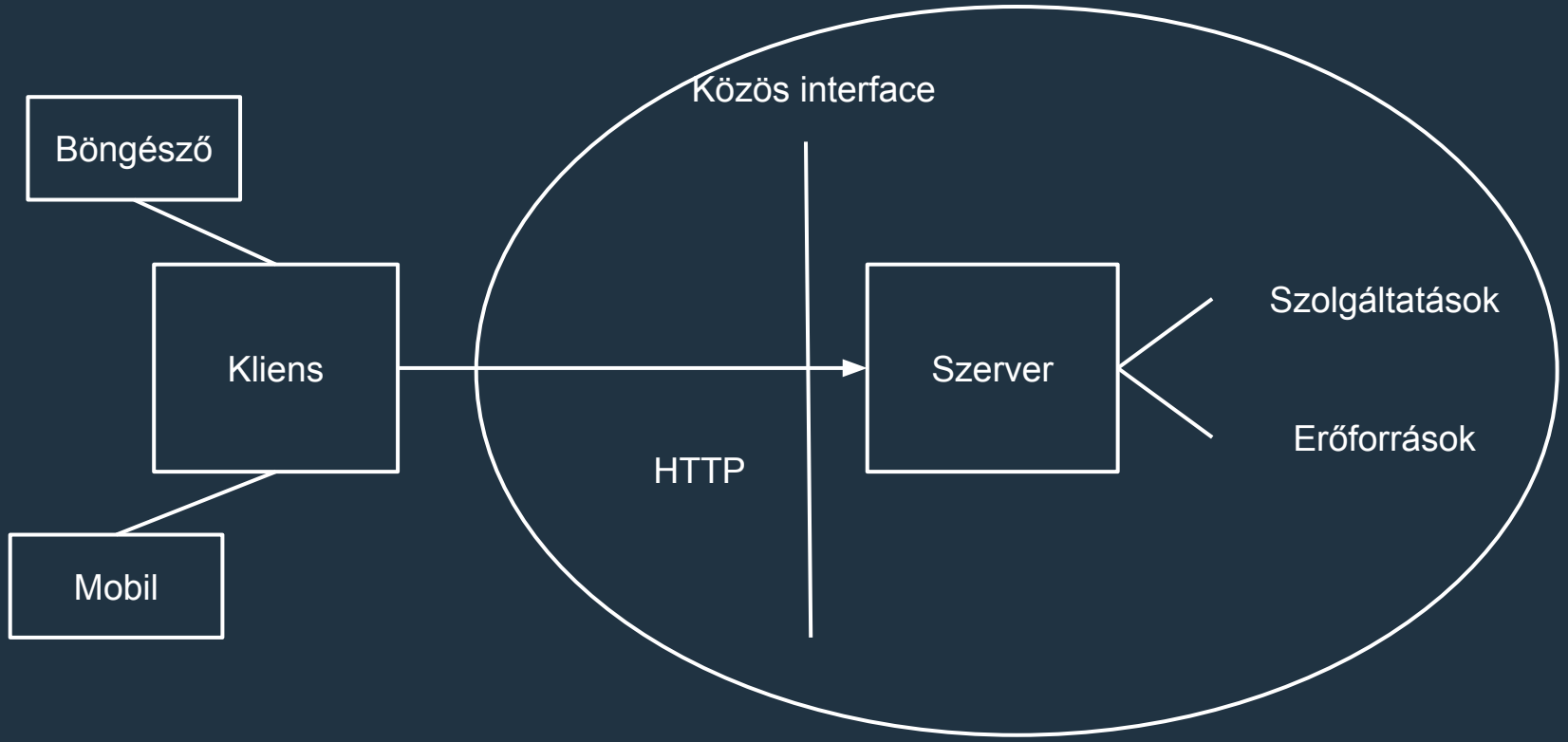
Miért HTTP?



- Frameworkok
  - Data marshaling
    - Library-k a HTTP kérések feldolgozáshoz

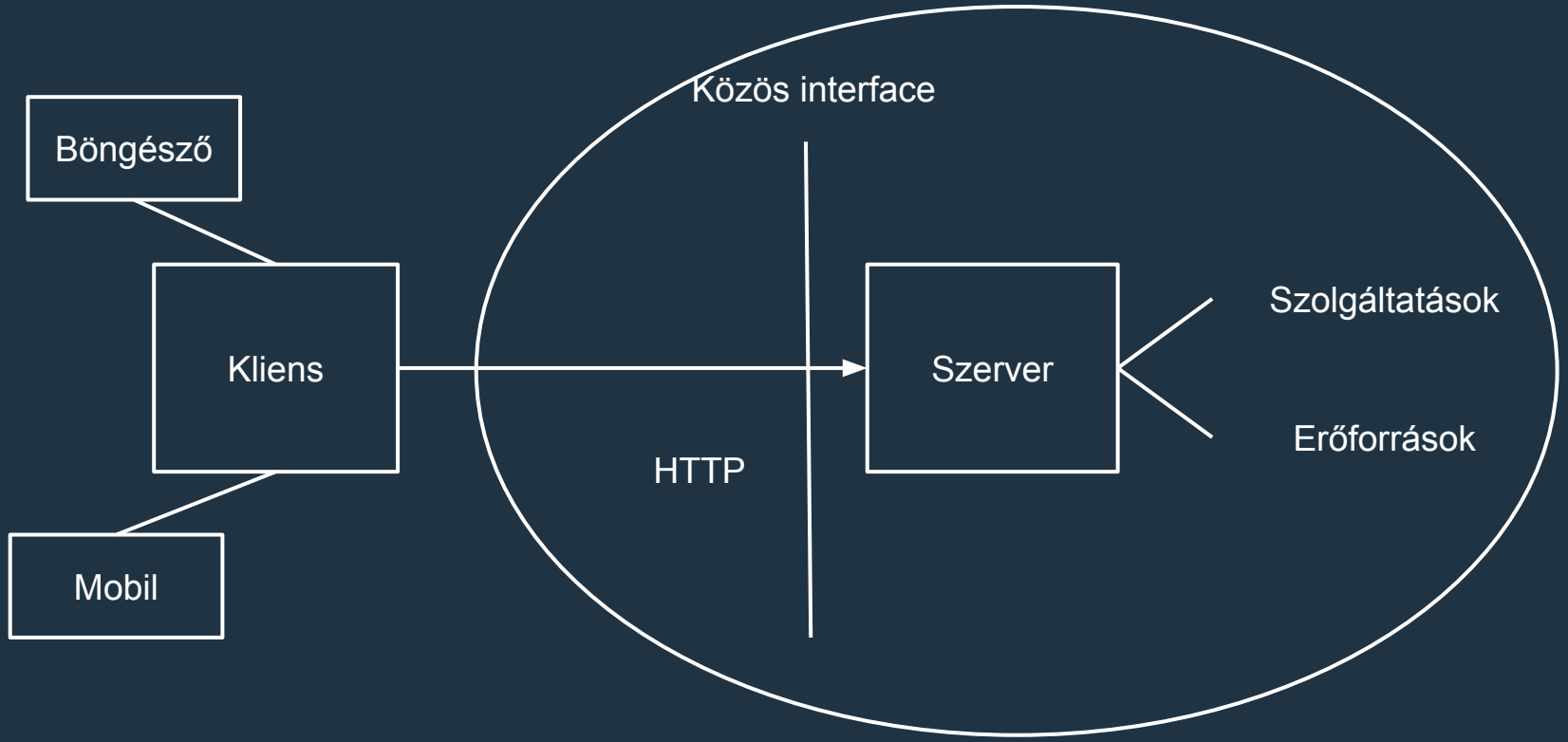
ERiSE

Miért HTTP?



- Frameworkok
- Data marshaling
- Library-k a HTTP kérések feldolgozáshoz
- Session kezelés

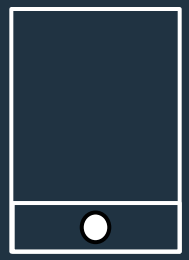
Miért HTTP?



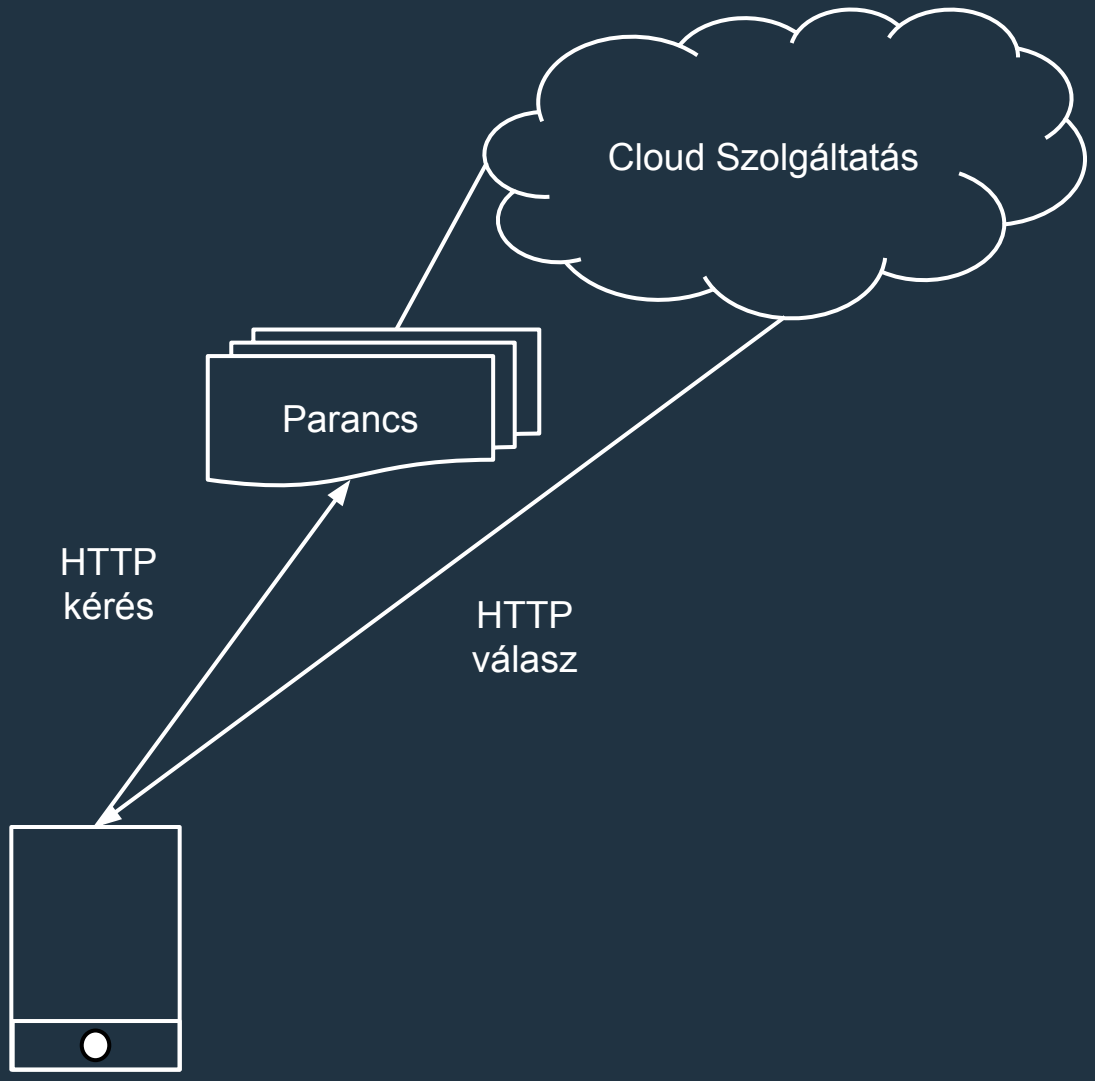
- Frameworkok
  - Data marshaling
    - Library-k a HTTP kérések feldolgozáshoz
  - Egyéb eszközök: Load Balancing
  - Session kezelés

ERiSE

Cloud szolgáltatás felépítése



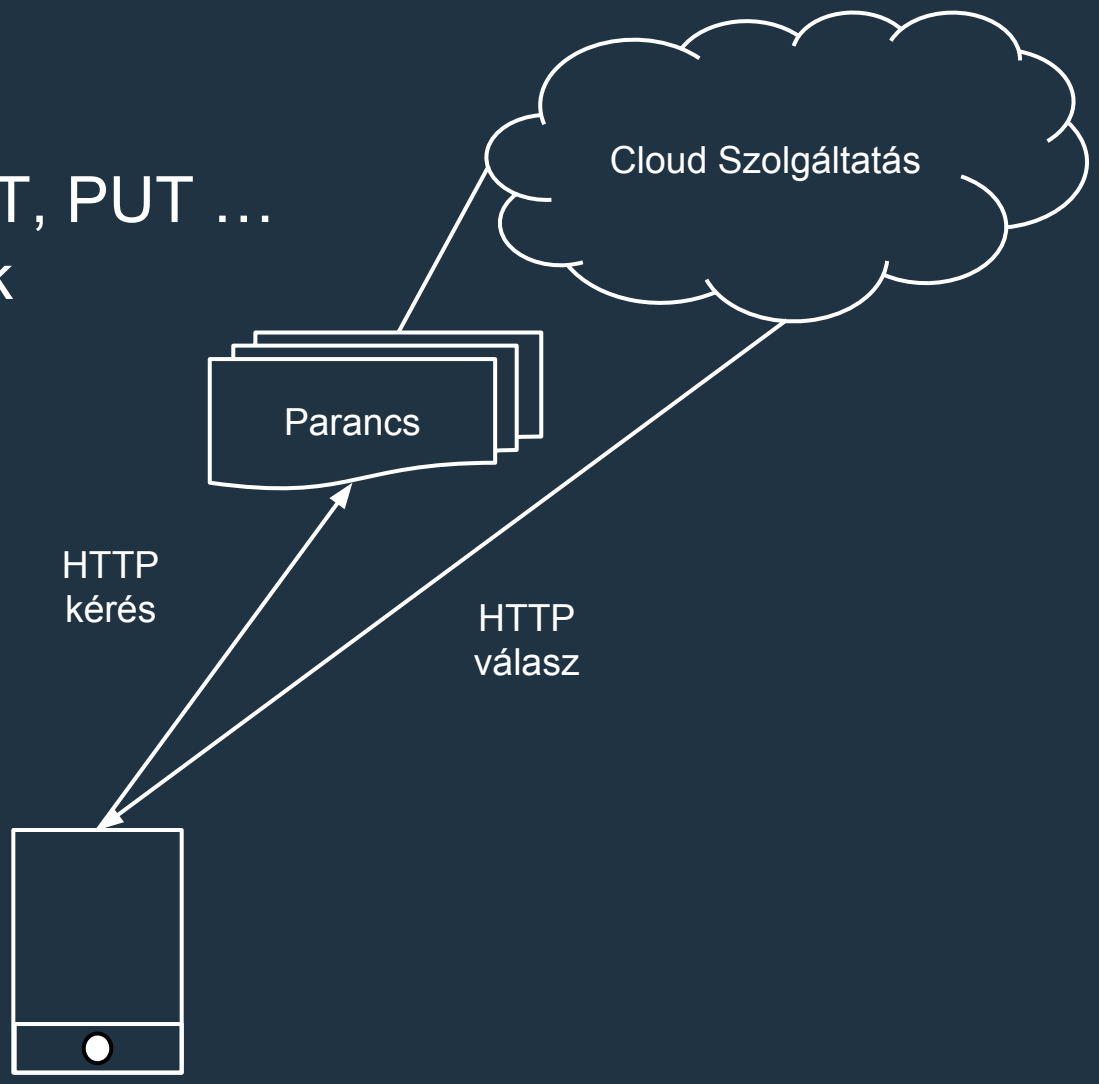
Cloud szolgáltatás felépítése



Cloud szolgáltatás felépítése

Kérés

- Method: GET, POST, PUT …
- Query praraméterek
- Content Type
- Body



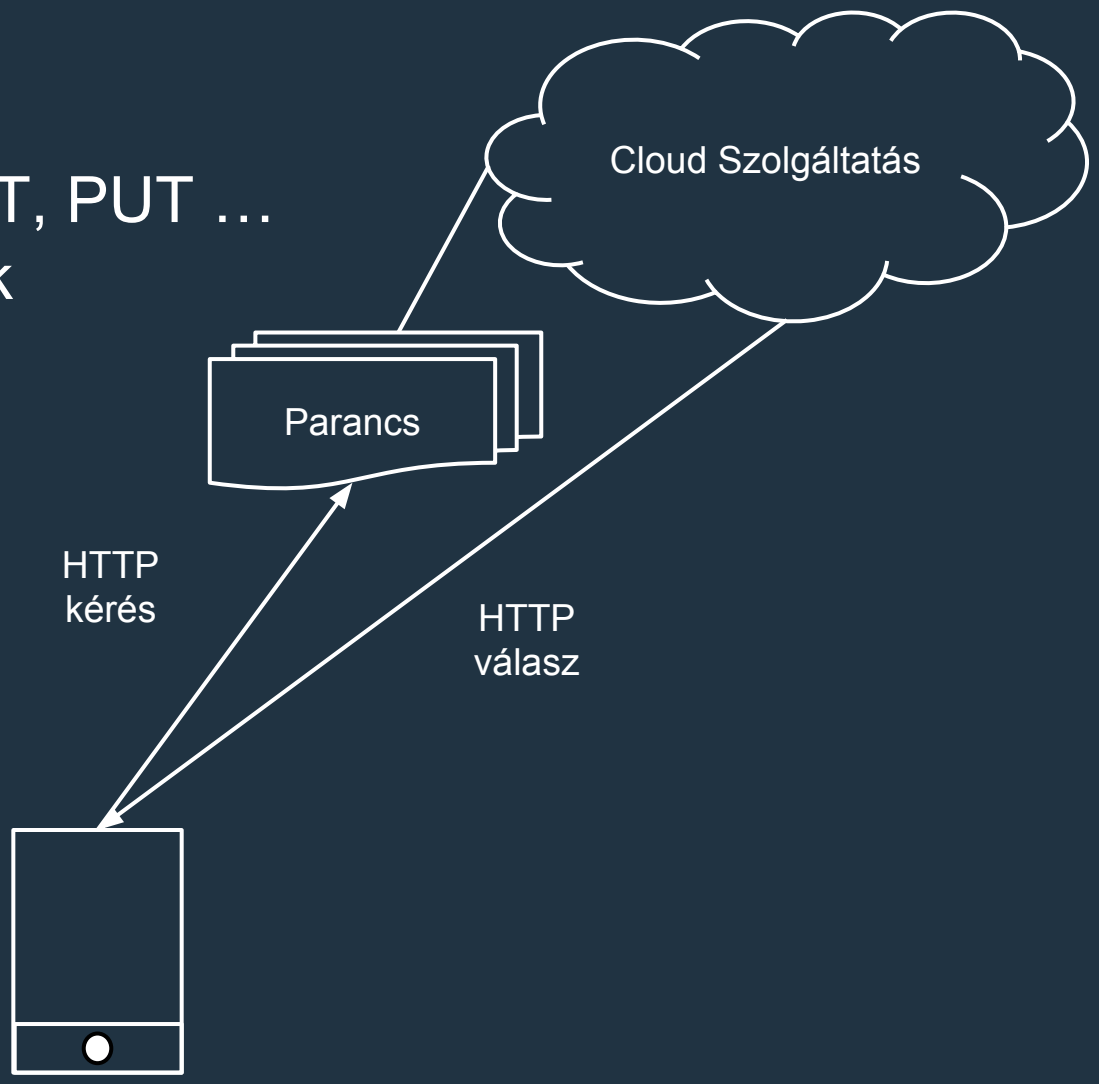
Cloud szolgáltatás felépítése

Kérés

- Method: GET, POST, PUT …
- Query praraméterek
- Content Type
- Body

Válasz

- HTTP Státusz kód
- Content Type
- Body



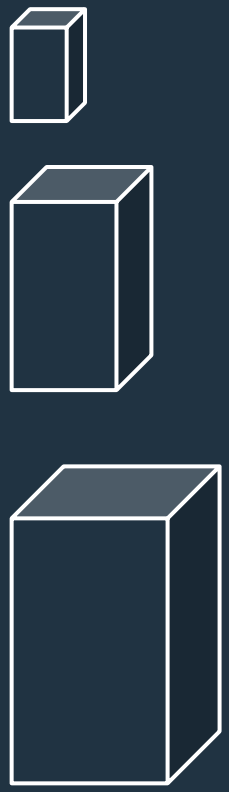
TERVEZÉSI / MEGVALÓSÍTÁSI KÉRDÉSEK

Skálázhatóság

Vertikális

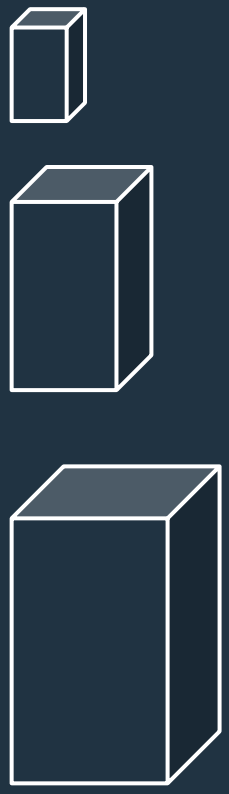
Skálázhatóság

Vertikális



Skálázhatóság

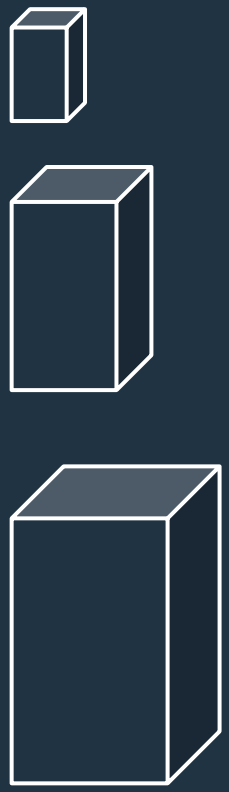
Vertikális



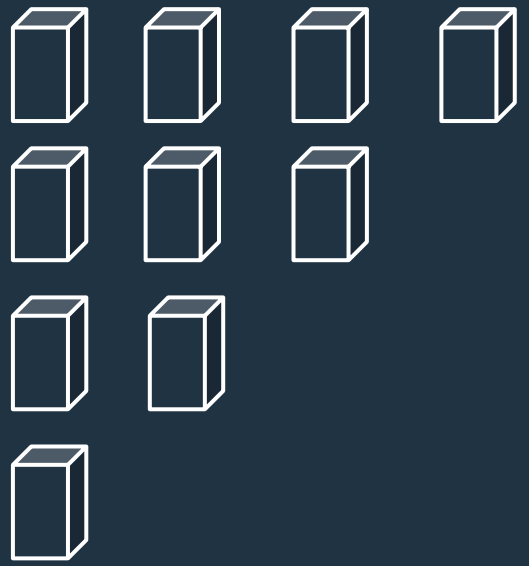
Horizontális

Skálázhatóság

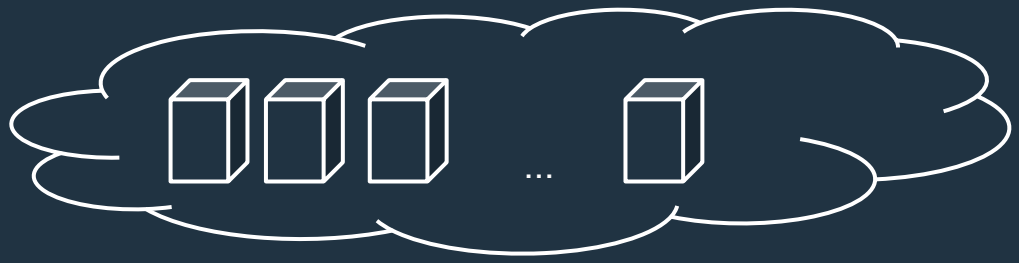
Vertikális



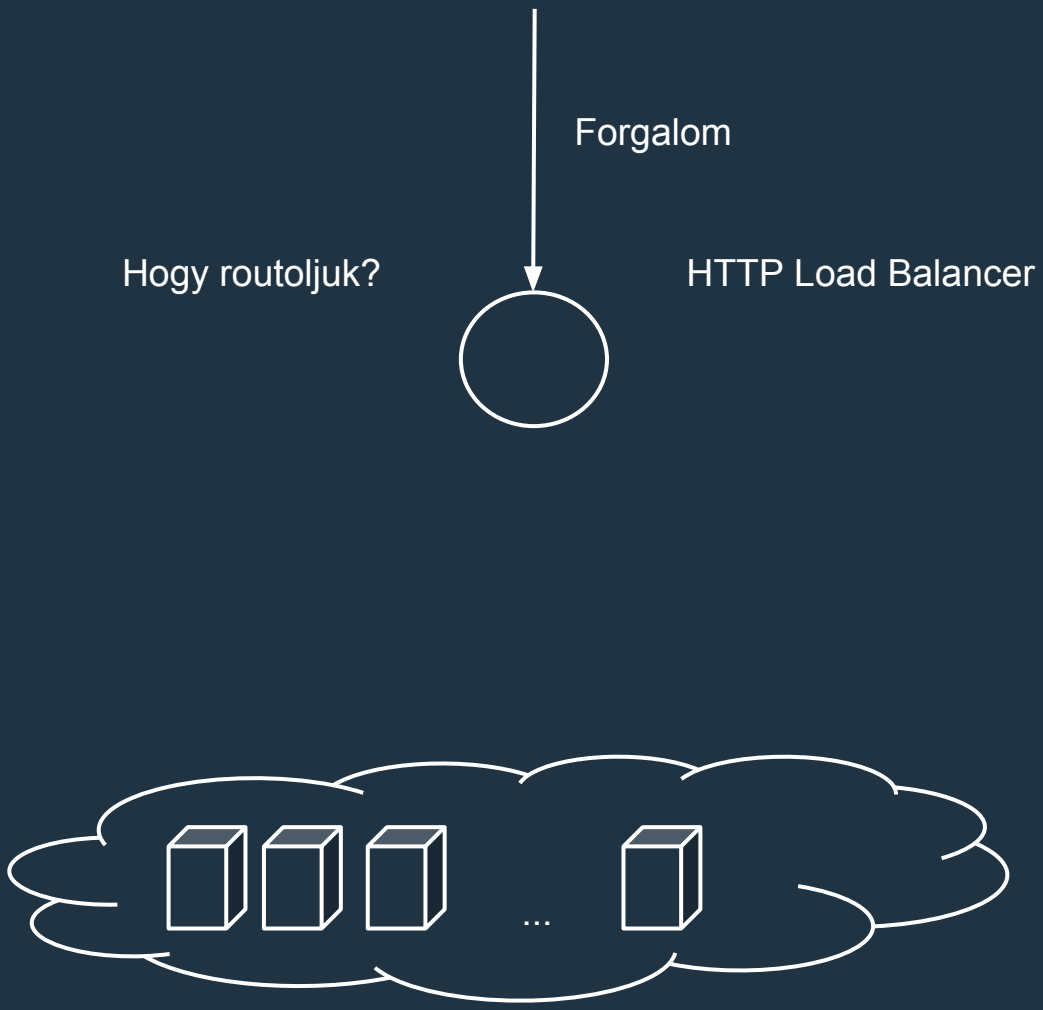
Horizontális



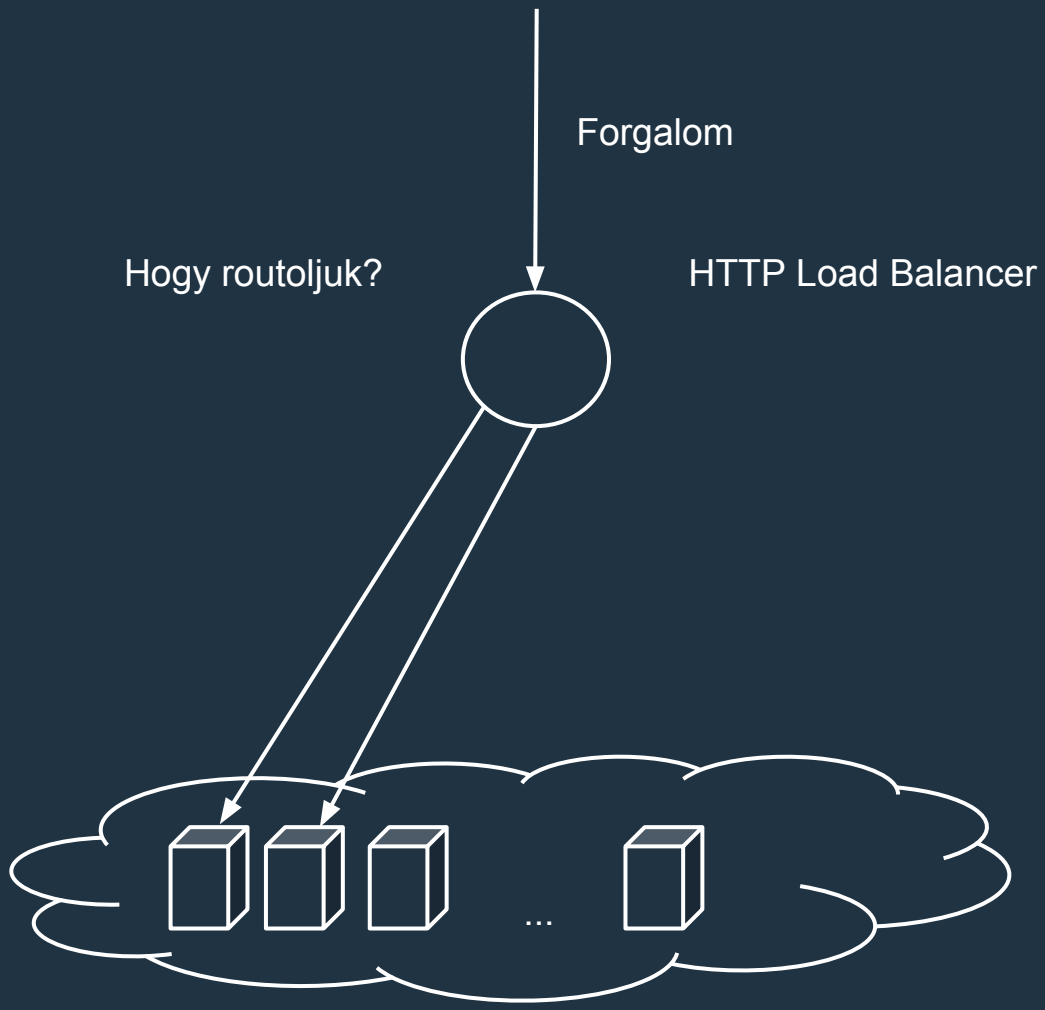
Loadbalancing 1/3



Loadbalancing 1/3

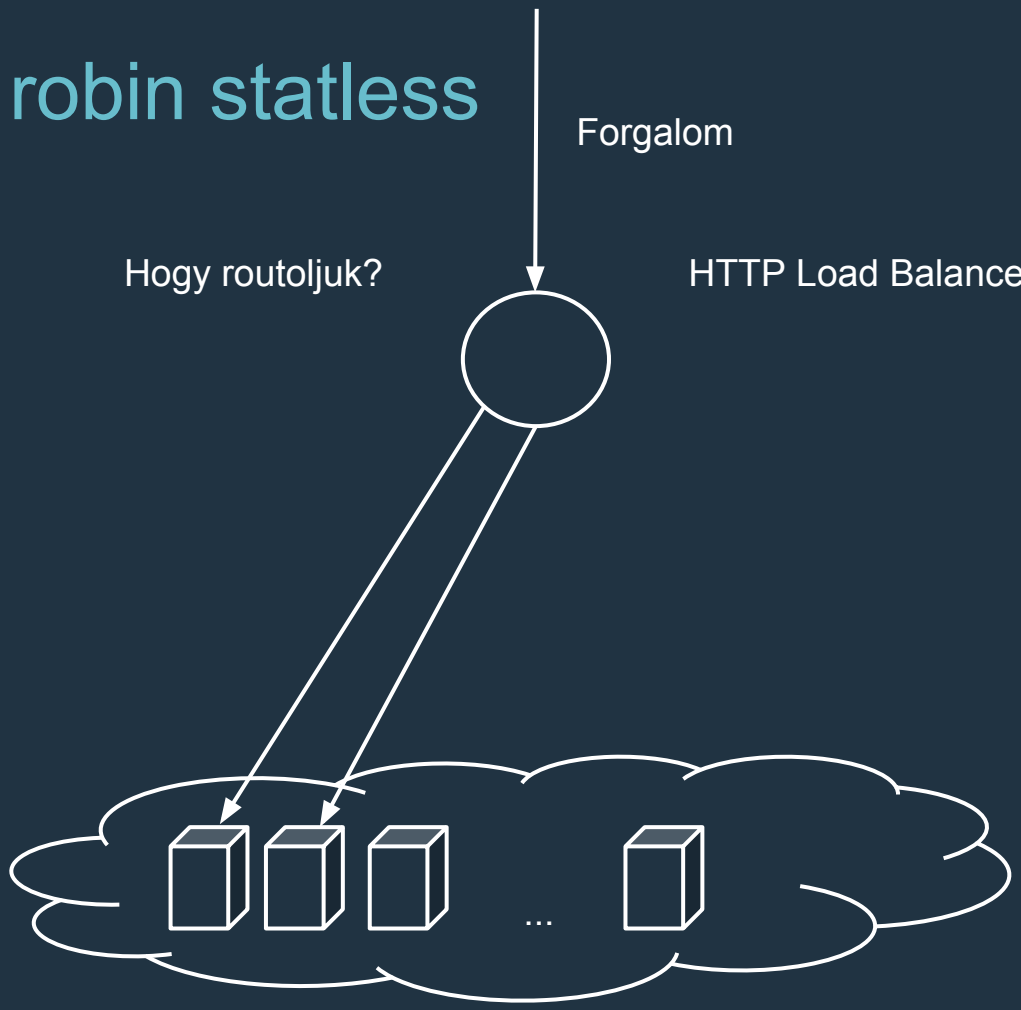


Loadbalancing 1/3

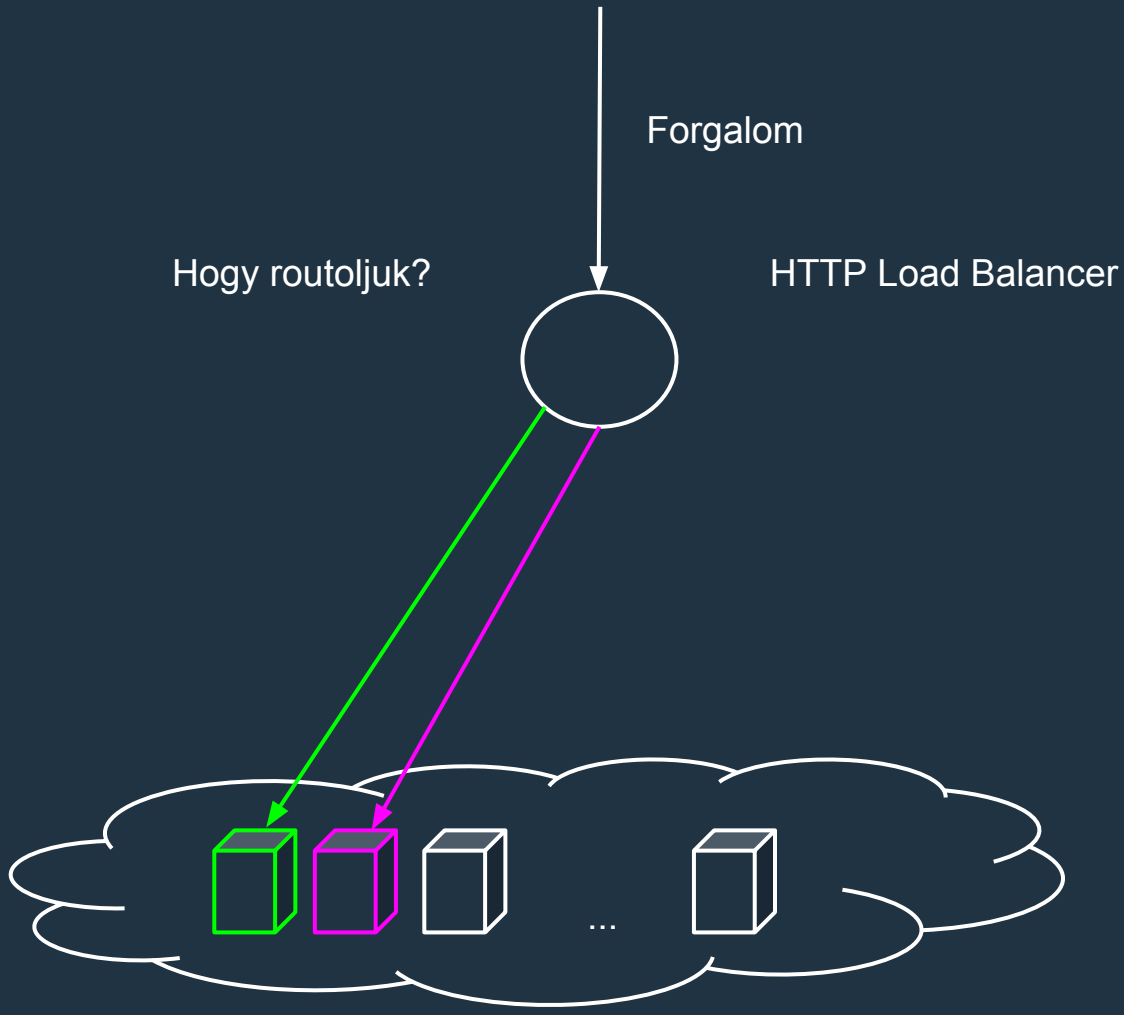


Loadbalancing 1/3

Round robin statless

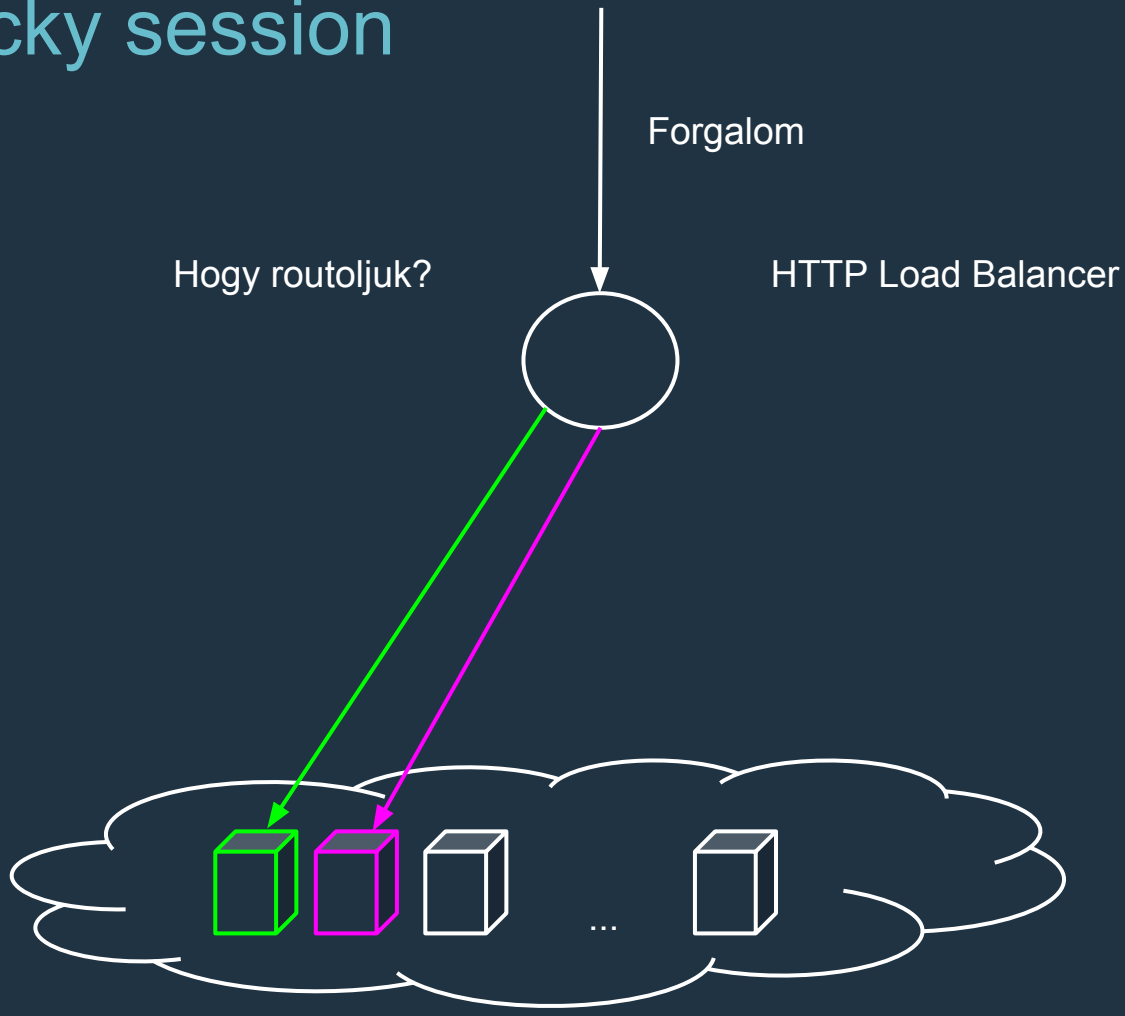


Loadbalancing 2/3

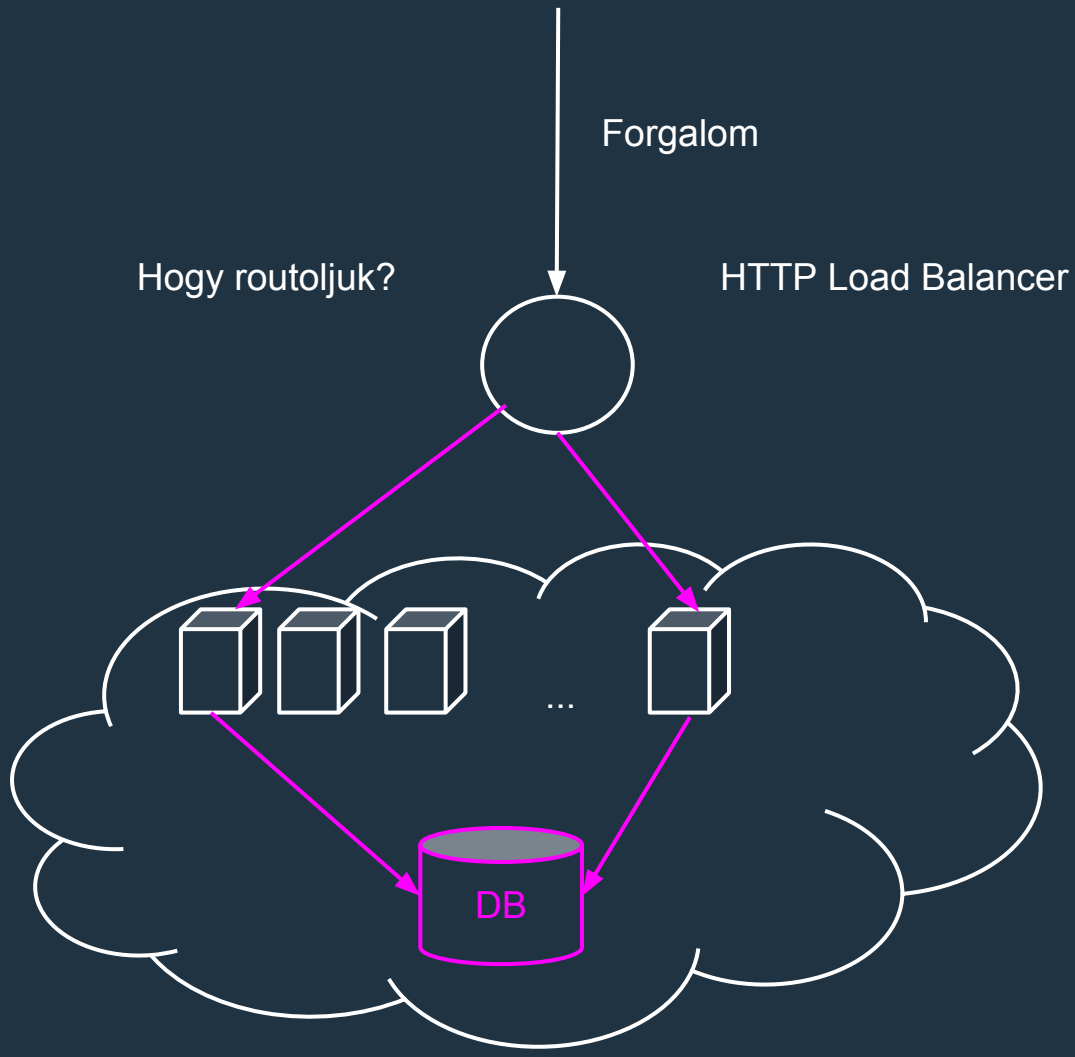


Loadbalancing 2/3

Sticky session

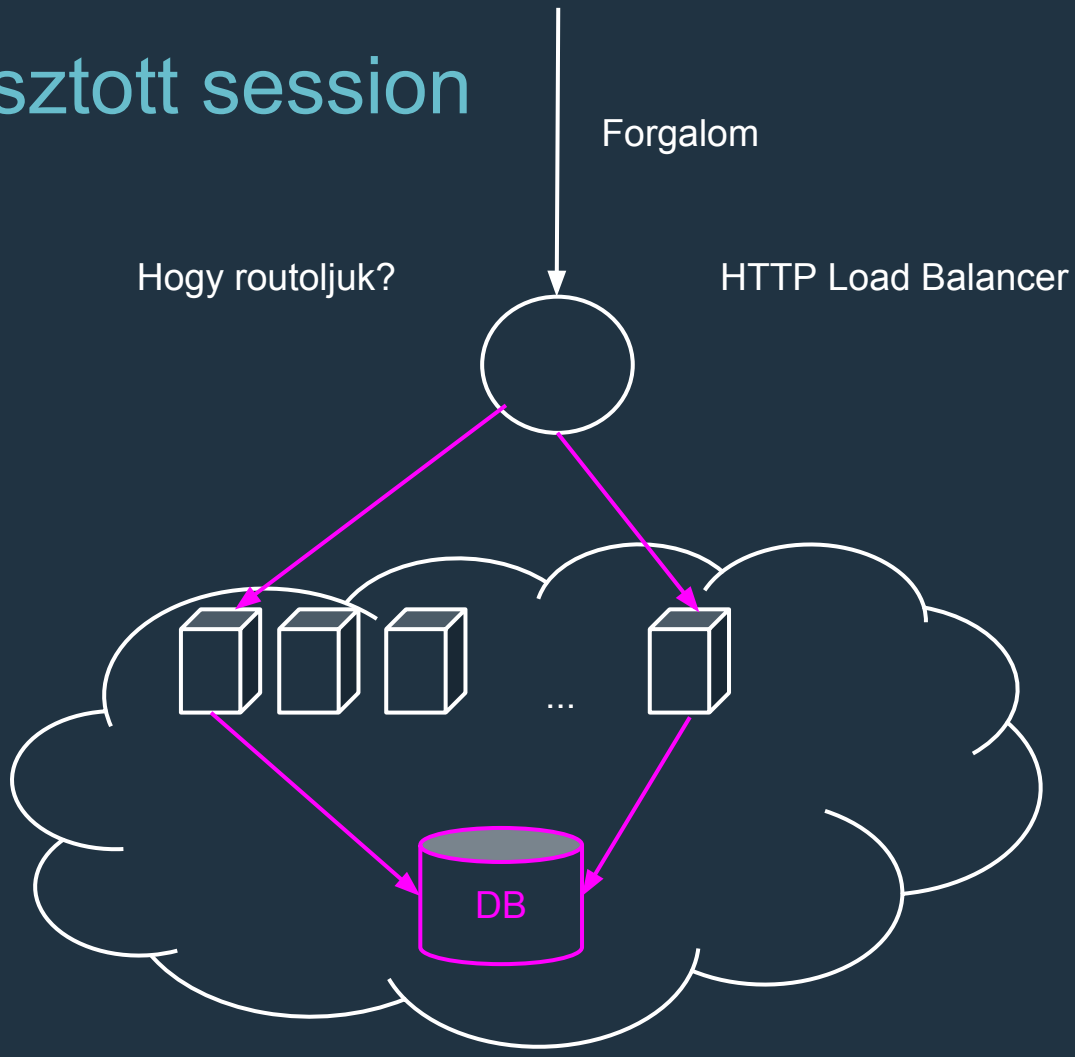


Loadbalancing 3/3

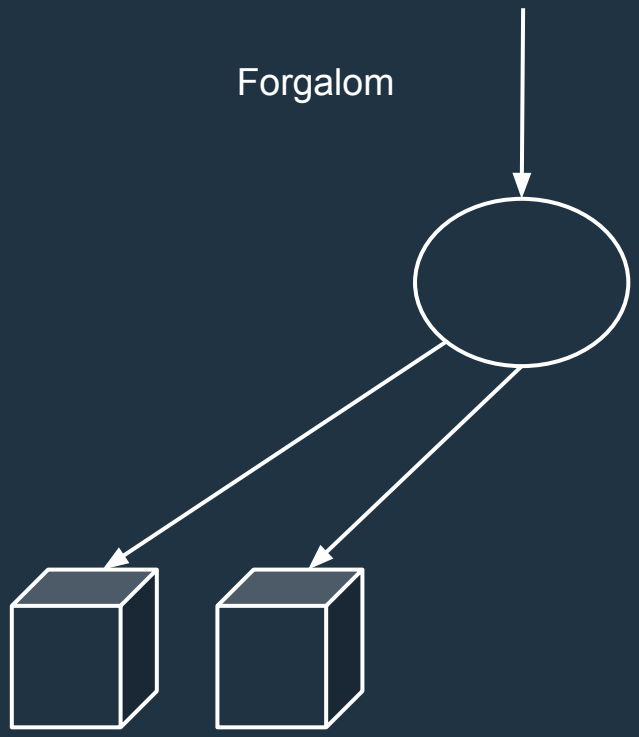


Loadbalancing 3/3

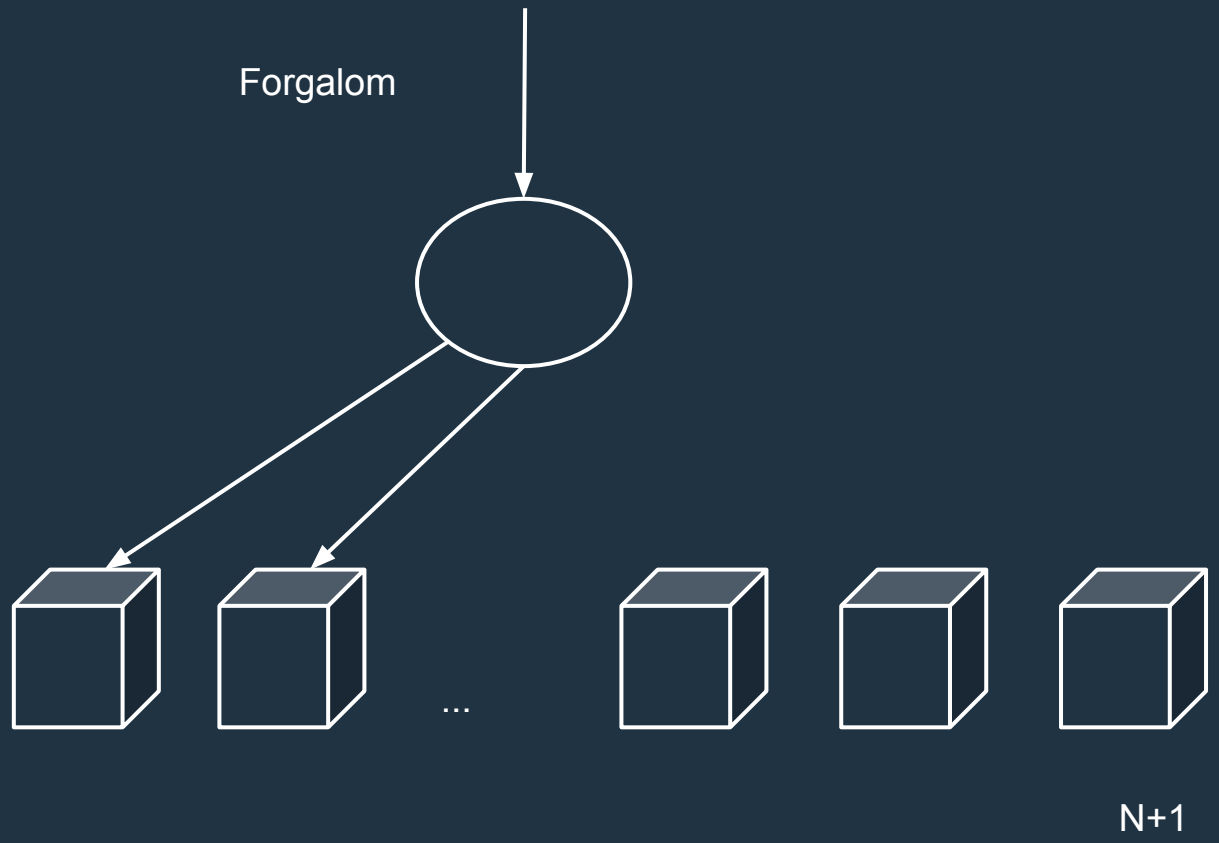
Elosztott session



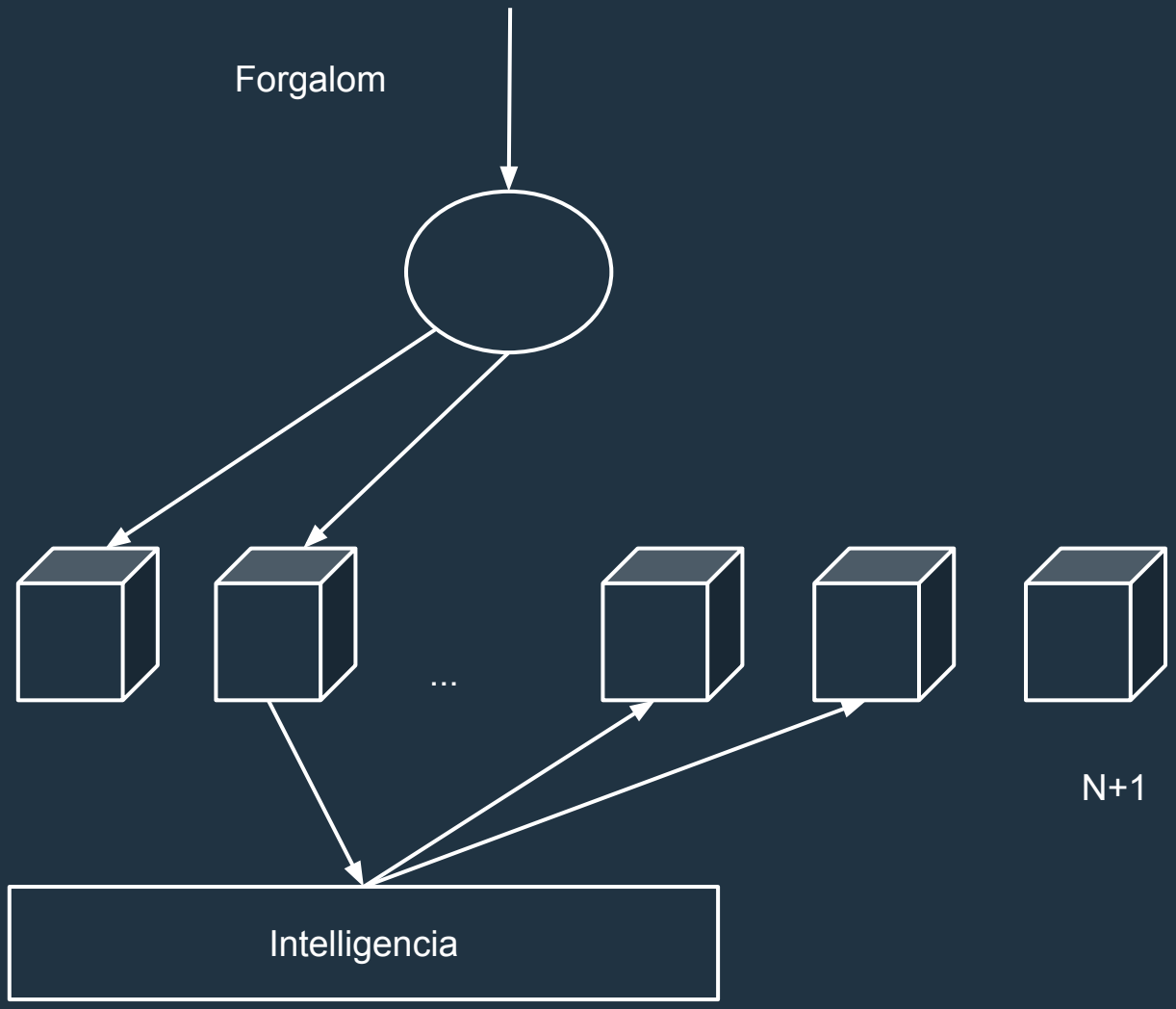
Autoscaling



Autoscaling



Autoscaling

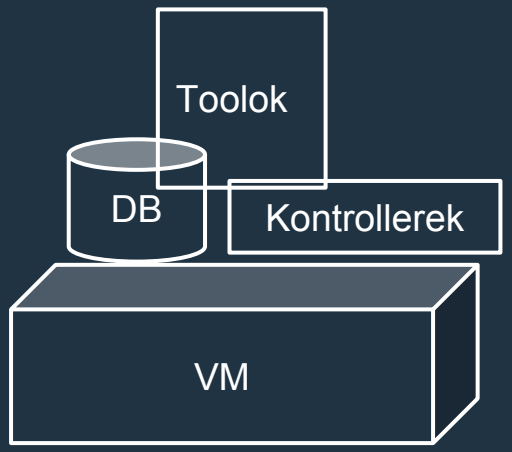


PaaS vs IaaS

IaaS

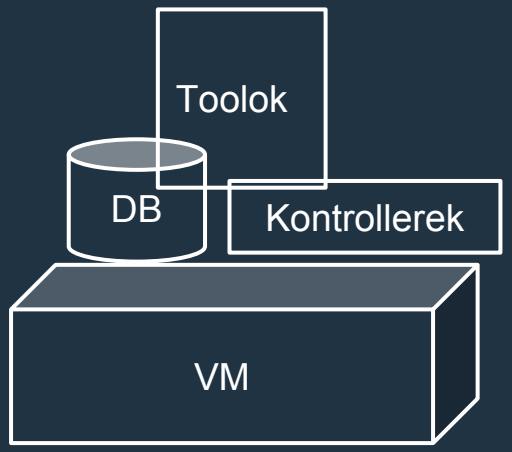
PaaS vs IaaS

IaaS



PaaS vs IaaS

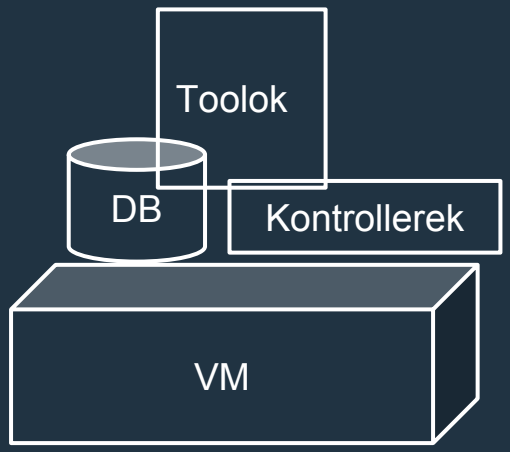
IaaS



Autoscaling

PaaS vs IaaS

IaaS



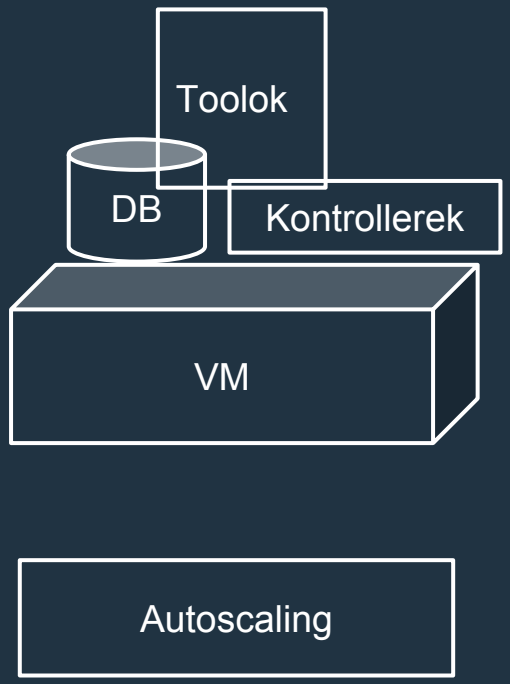
Autoscaling

PaaS

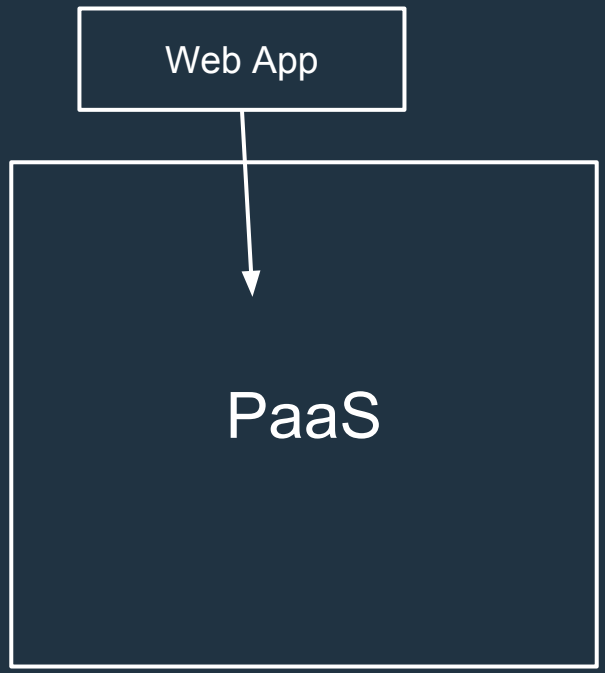
Web App

PaaS vs IaaS

IaaS



PaaS



AMAZON ELASTIC BEANSTALK

Mit tud egy PaaS

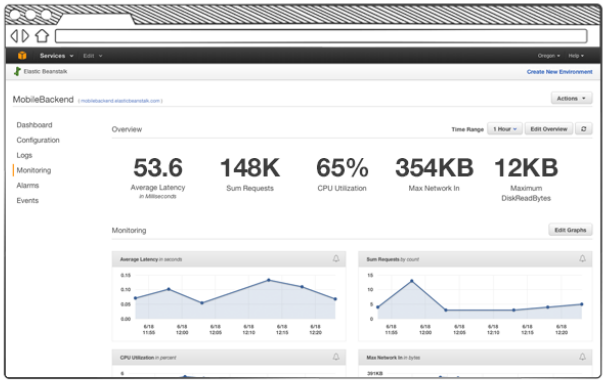
Pl.: Amazon Elastic Beanstalk

- Loadbalancing
- Auto scaling
- Multiplatform
  - .NET,
  - Node.js,
  - PHP,
  - Python,
  - Ruby,
  - Java
- Verzió kezelés
- Egyszerű deploy
- Egyszerű hozáférés a többi Amazon szolgáltatáshoz

ERiSE

Elastic Beanstalk

Alkalmazás létrehozás



Welcome to AWS Elastic Beanstalk

With Elastic Beanstalk, you can **deploy**, **monitor**, and **scale** an application quickly and easily. Let us do the heavy lifting so you can focus on your business.

To deploy your **existing web application**, create an application source bundle and then create a new application. If you're using **Git** and would prefer to use it with our command line tool, please see Getting Started with the EB CLI.

To deploy a **sample application** with just once click, select a platform and click **Launch Now**.

Docker — Looking for a different platform? Let us know.

Elastic Beanstalk will create an environment running Docker 1.0.0 on 64bit Amazon Linux 2014.03 v1.0.1.

Launch Now

Get Started in Three Easy Steps

Select a Platform

Upload an Application or Use a Sample

Run it!

Start Now by Selecting Your Platform

php · Ruby · Windows Server 2012 · node · ...and more

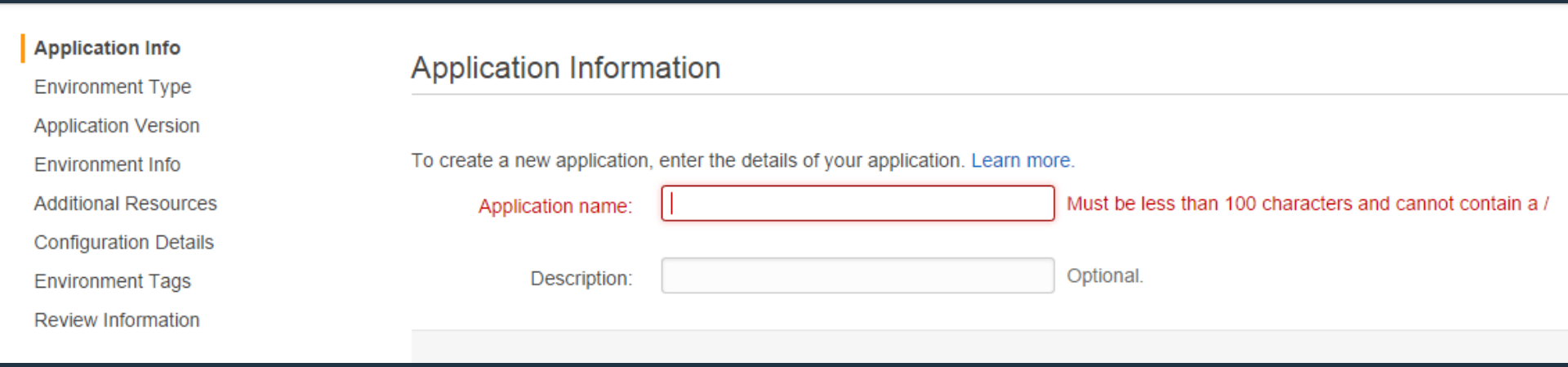
 Privacy Policy Terms of Use

Feedback

ERiSE

Elastic Beanstalk

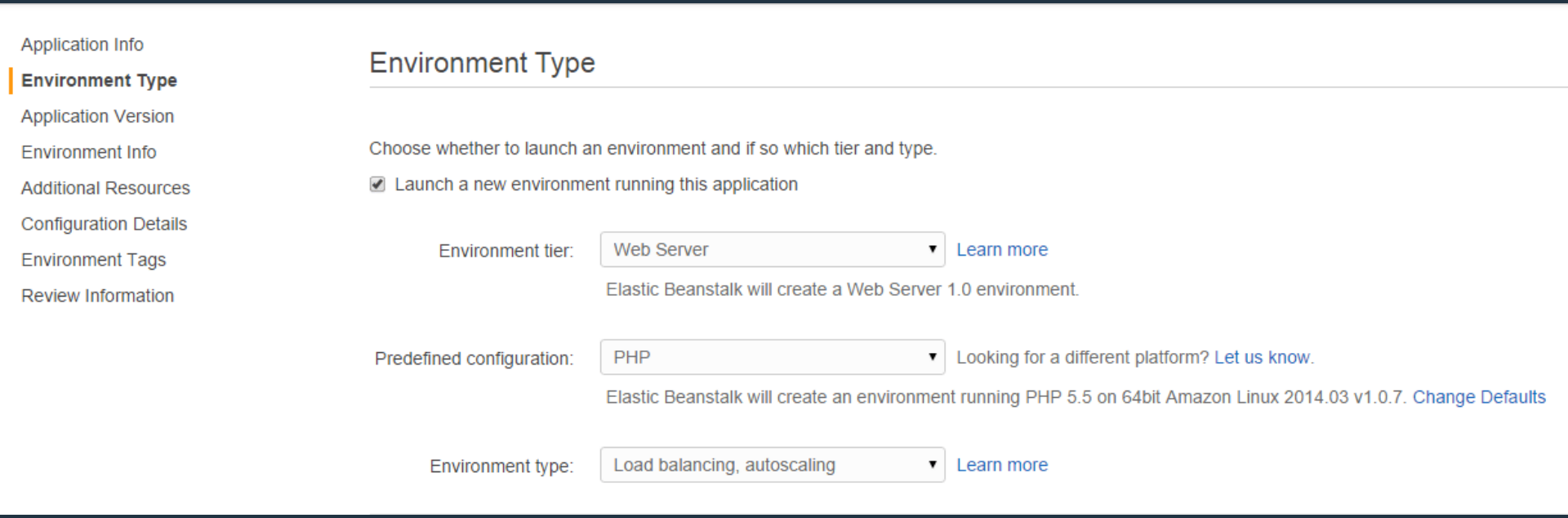
Meta adatok megadása



ERiSE

Elastic Beanstalk

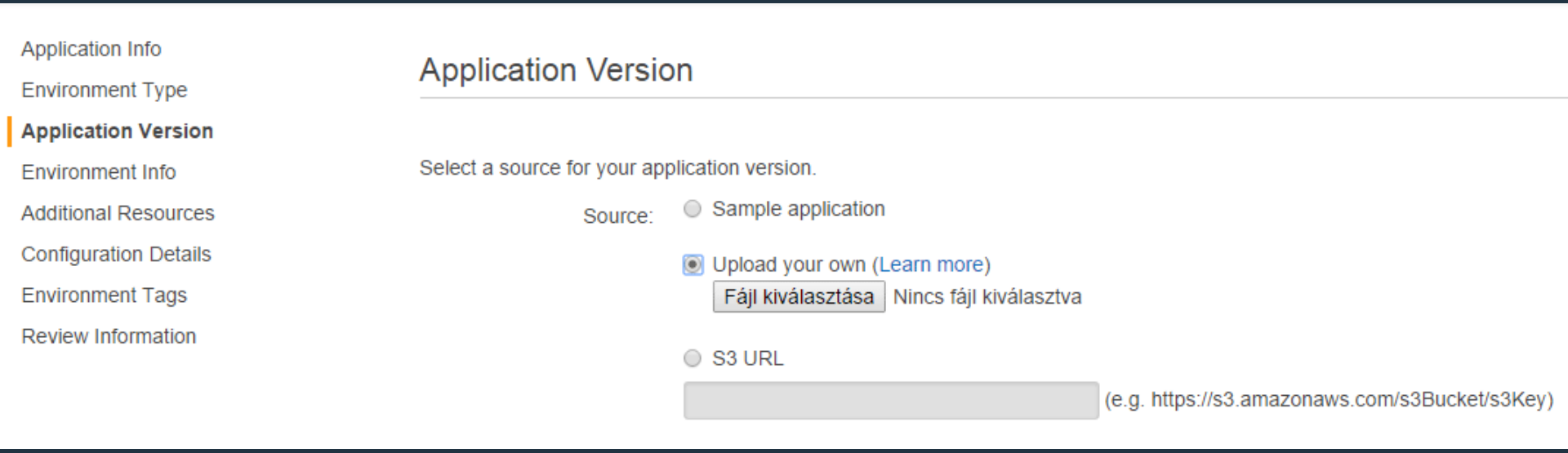
Alkalmazás típusának megadása



ERiSE

Elastic Beanstalk

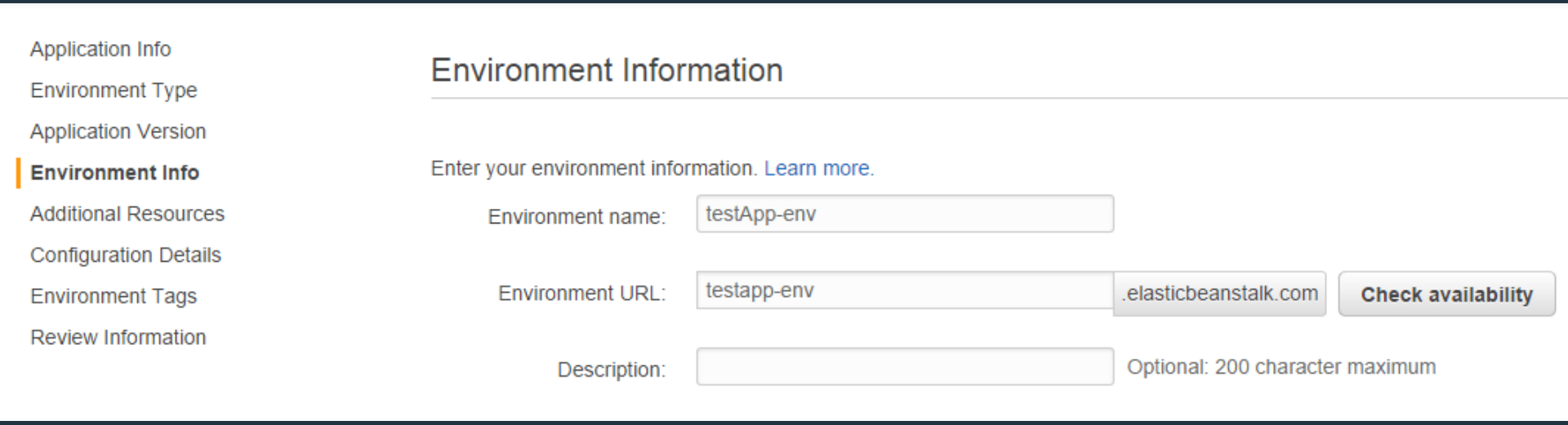
Forrás beállítása



ERiSE

Elastic Beanstalk

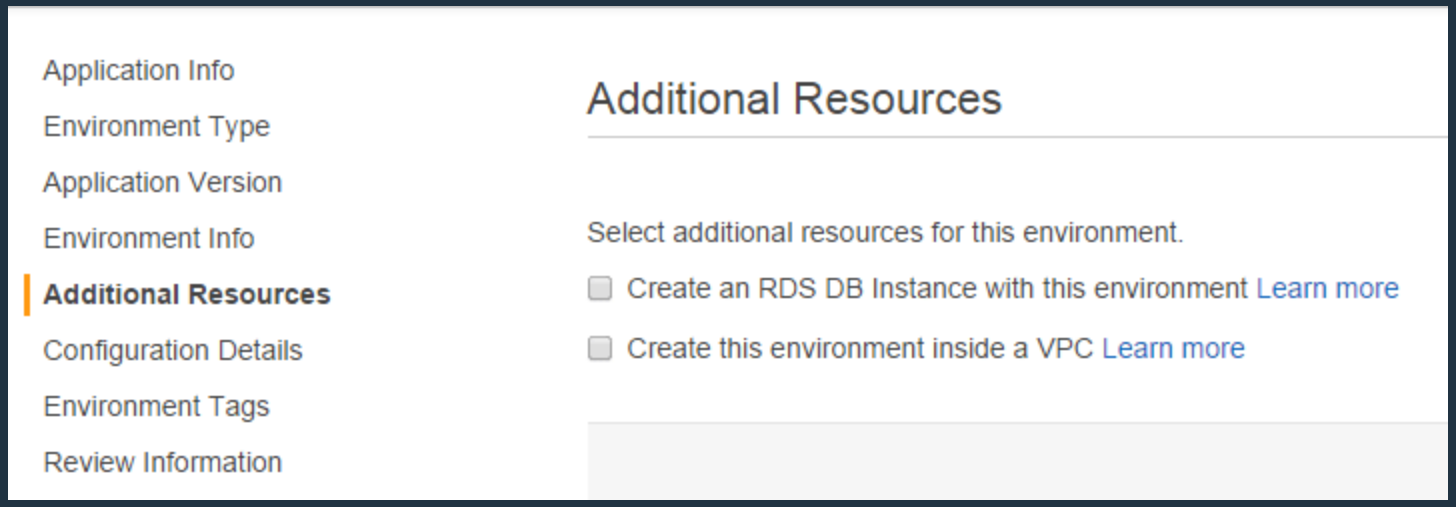
Elérhetőség és környezet beállítása



ERISE

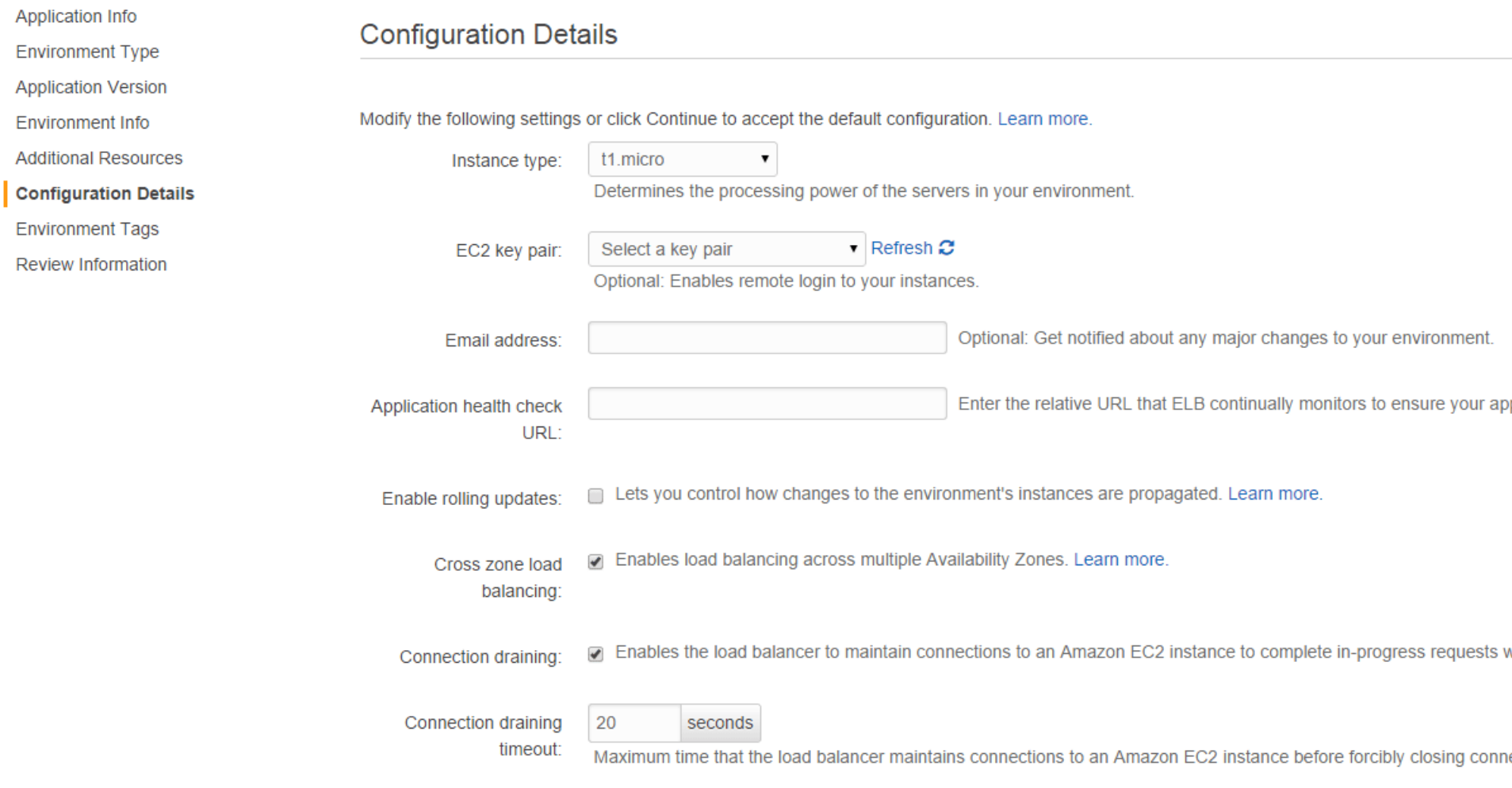
Elastic Beanstalk

Adatbázis és Virtual Private Cloud beállítás



Elastic Beanstalk

Egyéb beállítások



ERiSE

Elastic Beanstalk

Környezeti válltozók beállítása

ERiSE

Elastic Beanstalk

- Application Info
- Environment Type
- Application Version
- Environment Info
- Additional Resources
- Configuration Details
- Environment Tags
- **Review Information**

Review

Review the following information. Then click “Launch.”

Application Info

Application name	test-app
Description	test-app

Environment Type

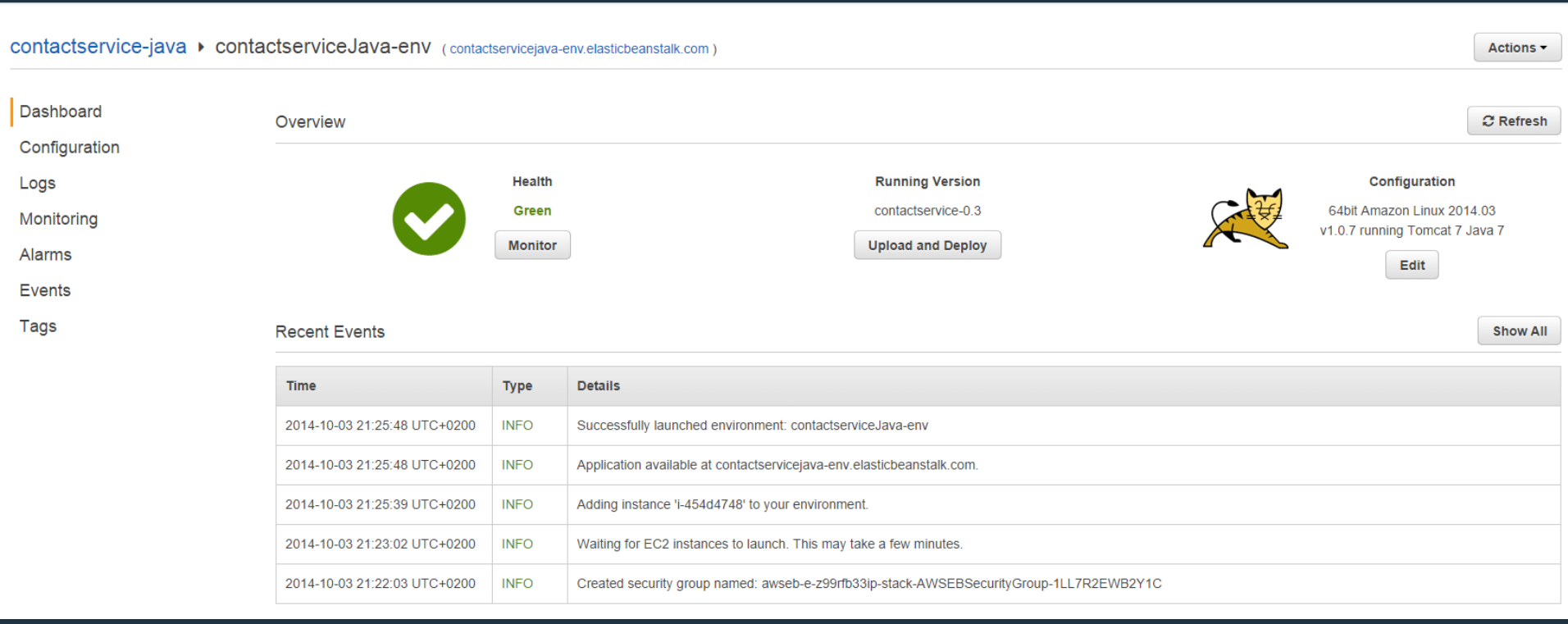
Tier	Web Server 1.0
Container type	64bit Amazon Linux 2014.03 v1.0.7 running PHP 5.5
Environment type	Load balancing, autoscaling

Application Version

Application source	Sample application

Elastic Beanstalk

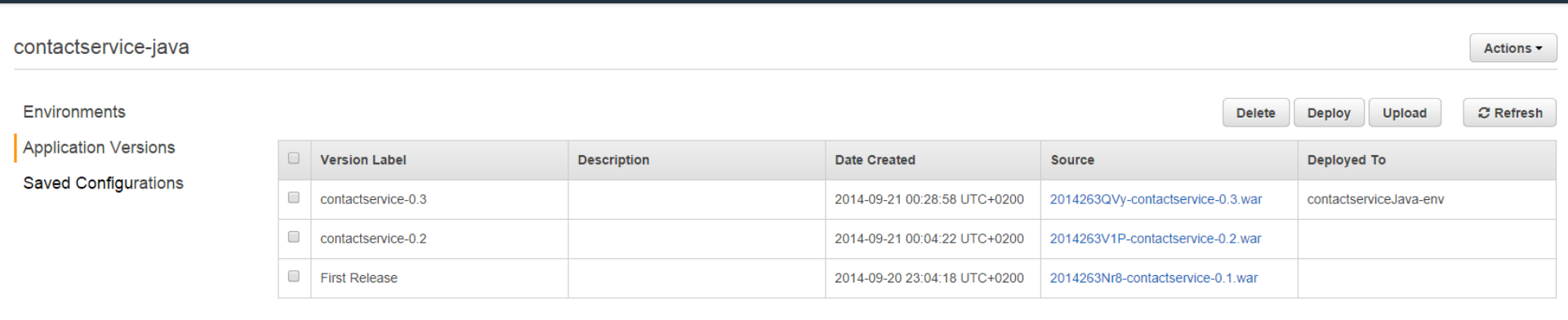
Alkalmazás dashboard



ERiSE

Elastic Beanstalk

Verziók kezelése



WEBSZOLGÁLTATÁSOK MEGVALÓSÍTÁSAI

Minta szolgáltatás

Kontakt szolgáltatás

- Kontaktok kezelése
  - Azonosító, Név, Email, Életkor,
- GET /contact - összes kontakt listája
- GET /contact/<id> - egy adott kontakt adatai
- POST /contact - kontakt létrehozása
- PUT /contact/<id> - kontakt módosítása
- DELET /contact/<id> - kontakt törlése

Dynamo BD

Amazon NoSQL Cloud adatbázis szolgáltatás

- Dokumentum és kulcs érték pár alapú adat model
- Háttérben SSD alapú adatárolás
- Nem tárterület, hanem sávszélesség alapú számlázás
- Séma felépítés:
  - Table - Item - Attributes
- Kulcsok:
  - Hash, Hash és Range típusú

Java

Spring

Java + Spring - Application


```
@EnableAutoConfiguration
@EnableDynamoDBRepositories
@EnableWebMvc
@ComponentScan
@Configuration
public class Application extends RepositoryRestMvcConfiguration {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }


}
```

Java + Spring Konfiguráció


```
@Value("${AWS_ACCESS_KEY_ID}")
private String amazonAWSAccessKey;

@Value("${AWS_SECRET_KEY}")
private String amazonAWSSecretKey;

@Bean
public AmazonDynamoDB amazonDynamoDB() {
    AmazonDynamoDB amazonDynamoDB = new AmazonDynamoDBClient(
            amazonAWSCredentials());
    return amazonDynamoDB;
}

@Bean
public AWSCredentials amazonAWSCredentials() {
    return new BasicAWSCredentials(amazonAWSAccessKey,
amazonAWSSecretKey);
}
```

Java + Spring Model


```
@DynamoDBTable(tableName = "Contacts")
public class Contact {

    private String id;
    private String name;
        //..

    @DynamoDBHashKey
    @DynamoDBAutoGeneratedKey
    public String getId() {
        return id;
    }

         // ...
    @DynamoDBAttribute
    public String getName() {
        return name;
    }

        //...
}
```

https://github.com/skromesch/webkonf2014-java

ERiSE

Java + Spring Crud Kontroller


```
@EnableScan
@RepositoryRestResource(path = ContactRepository.CONTACT_SVC_PATH)
public interface ContactRepository extends CrudRepository<Contact,
Long>{

    public static final String NAME_PARAMETER = "name";
    public static final String EMAIL_PARAMETER = "email";
    public static final String AGE_PARAMETER = "age";
    public static final String CONTACT_SVC_PATH = "/contact";

    public Collection<Contact> findByName(
            @Param(ContactRepository.NAME_PARAMETER) String name);
    public Collection<Contact> findByEmail(
            @Param(ContactRepository.EMAIL_PARAMETER) String
email);
    public Collection<Contact> findByAgeLessThan(
            @Param(ContactRepository.AGE_PARAMETER) int maxage);
}
```

Java + Spring Kontroller

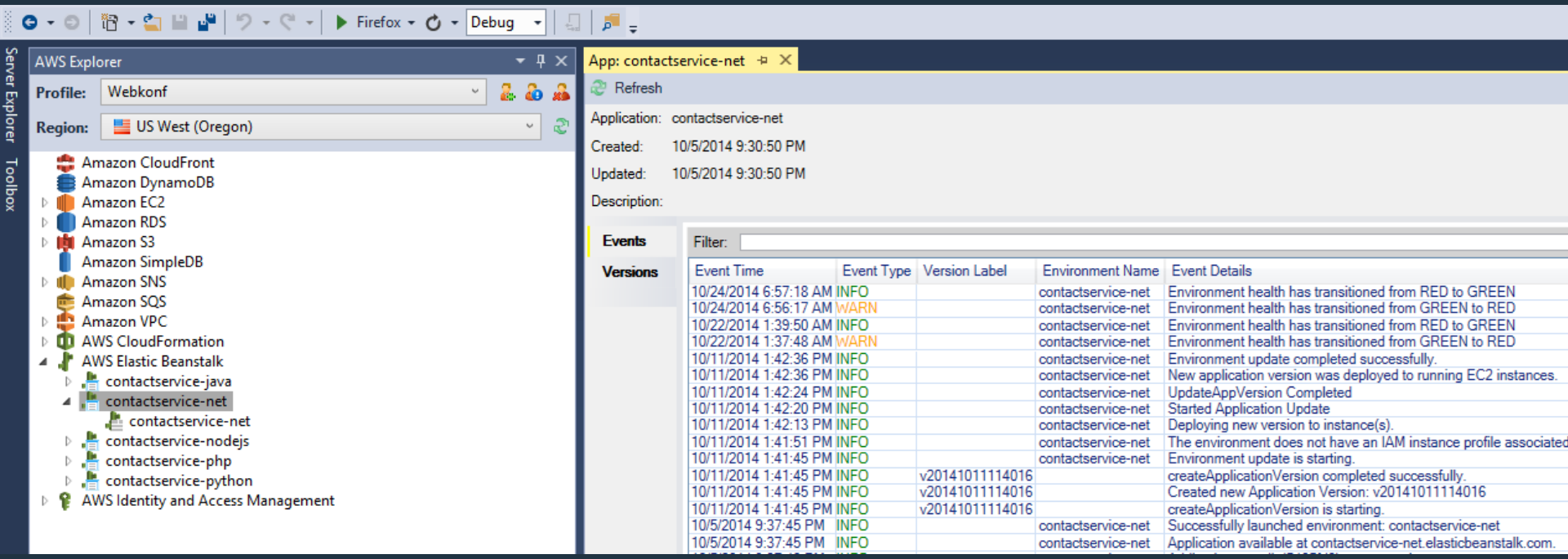

```
@Controller
public class CrudController {
    @RequestMapping( value="/crud/contact", method=RequestMethod.GET)
    public @ResponseBody Collection<Contact> getContactList() {
        return contacts.values();
    }

    @RequestMapping(value="/crud/contact",method=RequestMethod.POST)
    public @ResponseBody Contact addContact(
                @RequestBody Contact v) {
        save(v);
        return v;
    }


}
```

ASP.NET

MVC

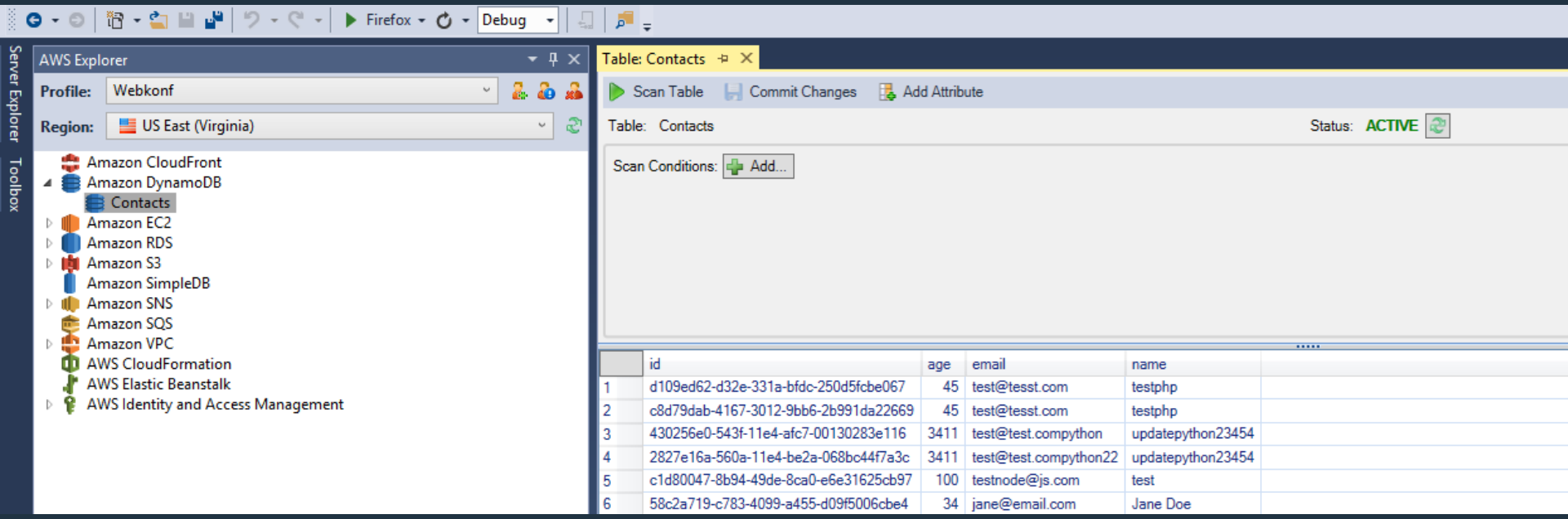
.NET VS Eszközök - AWS Explorer

Elastic Beanstalk



.NET VS Eszközök - AWS Explorer

DynamoDB



ERiSE

.NET + MVC Model


```
[DynamoDBTable("Contacts")]
public class Contact
{
    [DynamoDBHashKey(AttributeName="id")]
    public string Id { get; set; }

    [DynamoDBProperty(AttributeName = "email")]
    public string Email { get; set; }

    [DynamoDBProperty(AttributeName = "age")]
    public int Age { get; set; }

    [DynamoDBProperty(AttributeName = "name")]
    public string Name { get; set; }
}
```

.NET + MVC Kontroller


```
public class ContactController : ApiController
```

```
AmazonDynamoDBClient client = new AmazonDynamoDBClient();
```

```
<appSettings>
    <add key="AWSProfileName" value="Webkonf" />
    <add key="AWSRegion" value="us-east-1" />
    <add key="ClientSettingsProvider.ServiceUri"
value="" />
</appSettings>
```

```
config.Routes.MapHttpRoute(
                name: "DefaultApi",
                routeTemplate: "api/{controller}/{id}",
                defaults: new { id = RouteParameter.Optional }
            );
```

https://github.com/skromesch/webkonf2014-net

ERiSE

.NET + MVC GET Kérés


```
public IEnumerable<Contact> Get(){
    try {
        DynamoDBContext context = new DynamoDBContext(client);
        IEnumerable<Contact> contactQueryResults;
        contactQueryResults = context.Scan<Contact>(null);
        return contactQueryResults;
    }
    catch (Exception e) {
        throw new HttpResponseException(Request.CreateErrorResponse(
HttpStatusCode.InternalServerError, e.Message));
    }
}
```

.NET + MVC POST Kérés


```
public HttpResponseMessage Post([FromBody]Contact contact){
    try {
        DynamoDBContext context = new DynamoDBContext(client);
        context.Save<Contact>(contact);
        return new HttpResponseMessage(HttpStatusCode.OK);
    }

    catch (Exception e) {
       throw new HttpResponseException( Request.
CreateErrorResponse( HttpStatusCode.InternalServerError, e.Message));
    }
 }
```

Python

Flask

Python + Flask

Boto - AWS Elérés

```
if not boto.config.has_section('Credentials'):
    boto.config.add_section('Credentials')
    boto.config.set('Credentials', 'aws_access_key_id', str( os.
environ.get( 'AWS_ACCESS_KEY_ID' )))
    boto.config.set('Credentials', 'aws_secret_access_key', str(
os.environ.get( 'AWS_SECRET_KEY')))
```

Python + Flask GET Kérés


```
@application.route('/contact', methods=['GET'])
def get_all():
    try:
        contacts_table = Table('Contacts')
        contacts = contacts_table.scan()
        res = []
        for contact in contacts:
            res.append({ 'id':contact['id'], 'name':contact['name'],
'email':contact['email'],'age':str(contact['age'])})

        return Response(json.dumps(res),
mimetype='application/json')
    except Exception as e:
            return Response(json.dumps({'message' : str(e)}),
mimetype='application/json')
```

Python + Flask POST Kérés


```
@application.route('/contact', methods=['POST'])
def create():
    try:
        contacts_table = Table('Contacts')
        id = uuid.uuid1()
        name = str(request.form['name'])
        age = int(request.form['age'])
        email = str(request.form['email'])
        res = contacts_table.put_item(data={
            'id'    : str(id), 'name' : name,
            'email': email, 'age'   : age
            })
        if res == True:
            return Response(json.dumps({'message' : 'saved'}),
mimetype='application/json')
        else:
            return Response(json.dumps({'message' : 'notsaved'}),
mimetype='application/json')
    except Exception as e:
            return Response(json.dumps({'message' : str(e)}),
mimetype='application/json')
```

NODE.JS

Express

Node.js + Express Applikáció


```
app.get('/', routes.index);
app.get('/contact', contact.list);
app.get('/contact/:id', contact.getById);
app.post('/contact/', contact.create);
app.put('/contact/:id', contact.update);
app.delete('/contact/:id', contact.delete);
```

```
http.createServer(app).listen(app.get('port'), function(){
    console.log('Express server listening on port ' + app.get
  ('port'));
});
```

Node.js + Express Model

Vogels data mapper for AWS

https://github.com/ryanfitz/vogels

```
var vogels = require('vogels');
vogels.AWS.config.loadFromPath('./config.json');

var Contact = vogels.define('Contact', function (schema) {
    schema.UUID('id', {hashKey: true});
    schema.String('email');
    schema.String('name');
    schema.Number('age');
});

Contact.config({tableName: 'Contacts'});
```

Node.js + Express GET Kérés


```
exports.list = function(req, res, next){
   Contact.scan().exec(function(err,contacts){
     if (err) {
           res.statusCode = 500;
           res.json(err);
           return;
     }
     else {
       res.json(contacts);
     }

   });
};
```

Node.js + Express POST Kérés


```
exports.create = function(req, res){
   var email = req.param("email");
   if(email == null){
       res.statusCode = 400;
       res.json({"message":"Email parameter missing"});
       return;
   }
   //...
   Contact.create({email: email, name: name, age: age}, function (err,
  acc) {
     if (err) {
           res.statusCode = 500;
           res.json(err);
           return;
     }
     else {
           res.statusCode = 200;
           res.json({ message: 'Contact created' });
     }
   });
};
```

https://github.com/skromesch/webkonf2014-nodejs

ERiSE

PHP

ZF2

PHP + ZF2

AWS elérés ZF2-höz

https://github.com/aws/aws-sdk-php-zf2

Elérés kontrollerből:

```
protected function _getDynamoDbClient(){
    $aws    = $this->getServiceLocator()->get('aws');
    $client = $aws->get('dynamodb');
    return $client;

}
```

PHP + ZF2

JSON Válasz modul.config.php

```
'view_manager' => array( //Add this config
        'strategies' => array(
            'ViewJsonStrategy',
        ),
    ),
```

PHP + ZF2 Kontroller


```php
class ContactController extends AbstractRestfulController
{
    protected $collectionOptions = array('GET');
    protected $resourceOptions = array('GET','POST','PUT','DELETE');
    protected $tableName = "Contacts";

    public function getList(){    }

    public function get($id){    }

    public function delete($id){    }

    public function update($id,$data){    }

    public function create($data){    }
}
```

PHP + ZF2 GET Kérés


```php
public function get($id)
{
    try{
        $client = $this->_getDynamoDbClient();
        $response = $client->getItem(array(
            "TableName" => $this->tableName,
                "Key" => array(
                    "id" => array( Type::STRING => $id )
                )
            ));
        if(is_null($response["Item"])){
            return $this->_apiError(404,"Not found:".$id);
        }
        return new JsonModel($response["Item"]);
    }catch(\Exception $ex){
        return $this->_apiException($ex);
    }
}
```

PHP + ZF2 POST Kérés


```php
public function create($data){
    try{
        $client = $this->_getDynamoDbClient();
        if(!isset($data["name"])){
            return $this->_apiError(404,"Missing param name");
        }
        //...
        $response = $client->putItem(array(
            "TableName" => $this->tableName,
              "Item" => array(
                 "id"    => array(Type::STRING => $this->uuid()),
                 "name"  => array( Type::STRING => $data["name"] ),
                 "age"   => array( Type::NUMBER => $data["age"] ),
                 "email" => array( Type::STRING => $data["email"] )
            )));
        return new JsonModel(array("message" => "OK"));
    }catch(\Exception $ex){
        return $this->_apiException($ex);
    }
}
```

Összefoglalás

- Mikor használható?
  - Pl: nem áll rendelkezésre “üzemeltetési kompetencia”
- Mikor nem használható?
  - Pl: egyedi rendszer igény esetén
- Mire használható?
  - Pl: rapid élesítés, kb 10 perc alatt üzemképes
- Hogyan használható?
  - Feladathoz válasszunk eszközt/eszközöket

Köszönöm a figyelmet!

Kromesch Sándor
sandor.kromesch@erise.hu