

LATVIJAS UNIVERSITĀTE  
DATORIKAS FAKULTĀTE

**MĀKOŅSKAITĻOŠANAS PAKALPOJUMU  
IZMANTOŠANA MULTIREĢIONĀLU SISTĒMU  
UZTURĒŠANAI**

BAKALaura DARBS

Autors: **Eduards Egle**

Studenta apliecības Nr.: ee13014

Darba vadītājs: M. dat. Andris Ševčenko

RĪGA 2017

## ANOTĀCIJA

Bakalaura darba “Mākoņskaitļošanas pakalpojumu izmantošana multireģionālu sistēmu uzturēšanai” mērķis ir izpētīt, kas ir multireģionālas sistēmas un kāda loma mākoņskaitļošanai ir šādu sistēmu uzturēšanā. Pētījumā tiek apskatīti un salīdzināti 3 mākoņskaitļošanas pakalpojumu sniedzēji, kā arī tiek veikta gadījuma izpēte sistēmai, kurā tiek risinātas problēmas, kas saistītas ar multireģionālas sistēmas izstrādi un uzturēšanu.

Atslēgvārdi: multireģionāls, informācijas sistēma, mākoņskaitļošana, virtuālā mašīna.

## **ABSTRACT**

### CLOUD COMPUTING IN MULTIREGIONAL SYSTEM HOSTING

Bachelor thesis “Cloud computing in multiregional system hosting” purpose is to research multiregional systems and how cloud computing can aid in developing and maintaining them. In this research author compares 3 different cloud computing service providers and does a case study of a system, which deals with problems concerning multiregional system development and maintenance.

Keywords: multiregional, data system, cloud computing, virtual machine.

# SATURS

APZĪMĒJUMU SARAKSTS .....	6
IEVADS .....	7
1. MĀKOŅSKAITĻOŠANA .....	9
1.1. Mākoņskaitļošanas jēdziens un vēsture .....	9
1.2. Mākoņskaitļošanas pakalpojumu līmeņi .....	10
1.2.1. Infrastruktūra kā serviss .....	11
1.2.2. Platforma kā serviss.....	12
1.2.3. Programmatūra kā serviss.....	13
2. MĀKOŅSKAITĻOŠANAS SERVISU DATU LOKĀCIJAS JURIDISKIE IEROBEŽOJUMI .....	14
2.1. Datu aizsardzības likumi .....	14
2.2. Valdības uzraudzība .....	15
2.3. Jurisdikcijas nozīme mākoņskaitļošanā .....	16
3. MĀKOŅSKAITĻOŠANAS PAKALPOJUMU SNIEDZĒJU SALĪDZINĀJUMS MULTIREGIONĀLĀS INFORMĀCIJAS SISTĒMĀS .....	18
3.1. Mākoņskaitļošanas pakalpojumu sniedzēji .....	18
3.1.1. Amazon Web Services .....	18
3.1.2. Microsoft Azure .....	19
3.1.3. Google Cloud Platform .....	19
3.2. Mākoņskaitļošanas pakalpojumu sniedzēju salīdzinājuma kopsavilkums.....	19
3.3. Datu centru lokācijas.....	22
3.4. Komūnas atbalsts .....	23
3.5. Monitoringa rīki .....	24
3.6. Domēnu nosaukumu pakalpojumi.....	25
3.7. Automātiskas mērogojamības iespējas .....	26
3.8. Automātiska izvietošana .....	27
3.9. Pakalpojuma izmaksas .....	29
3.10. Tehniskais atbalsts .....	30
3.11. Slodzes līdzsvarošanas iespējas .....	32
3.12. Mākoņglabāšana.....	33
4. GADĪJUMA IZPĒTE - ZIEDOJUMU PIESAISTES PLATFORMA <i>FUNDERFUL</i> .....	36
4.1. <i>Funderful</i> ziedojumu piesaistes platforma.....	36
4.2. Mākoņskaitļošanas pakalpojumu sniedzēja maiņas pamatojums .....	37

4.3. Jaunās <i>Funderful</i> sistēmas izstrāde un infrastruktūra .....	39
4.4. Sistēmas arhitektūra .....	39
4.5. Reģionu datu bāžu datu sinhronizācija .....	40
4.6. Aizķeru implementācija multireģionālā sistēmā .....	41
4.7. Multireģionālu sistēmu automātiska izvietošana .....	43
4.8. Gadījuma izpētes kopsavilkums .....	45
REZULTĀTI.....	46
SECINĀJUMI.....	47
IZMANTOTĀ LITERATŪRA UN AVOTI .....	49
PIELIKUMI .....	55
1. pielikums. Risinājuma pirmkods reģionu datu bāžu savienojumu iegūšanai .....	55
2. pielikums. Risinājuma pirmkods reģionu datu sinhronizācijai.....	56
3. pielikums. Risinājuma pirmkods aizķeru apstrādei multireģionālā sistēmā.....	57
4. pielikums. Sistēmas automātiskās izmitināšanas programmkoda fragmenti.....	59

## APZĪMĒJUMU SARAKSTS

**AWS (Amazon Web Services)** – mākoņskaitļošanas pakalpojumu sniedzējs

**DNS (Domain Name System)** – interneta serviss, ar kuru domēnu nosaukumi tiek tulkoti uz IP adresēm

**EC2 (Elastic Compute Cloud)** – *AWS* piedāvātais mākoņskaitļošanas pakalpojums

**FTP/FTPS** – tīmekļa protokols, kurš izmantojams datņu pārvietošanai tīklā no servera uz klientu, izmantojot klienta-servera modeli

**Google Cloud Pub/Sub** – reāllaika ziņojumu serviss, kurš ļauj sūtīt un saņemt informāciju starp neatkarīgām lietotnēm

**IP (Internet Protocol)** – metode vai protokols, ar kuru interneta vidē tiek sūtīti dati no vienas ierīces uz citu

**IaaS (Infrastructure as a Service)** – mākoņskaitļošanas pakalpojumu veids, kur tīmekļa vidē tiek piedāvāti virtuāli skaitļošanas resursi

**PaaS (Platform as a Service)** – mākoņskaitļošanas kategorija, kurā izstrādātājiem tiek piedāvāta platforma un vide, lai izstrādātu lietotnes un piedāvātu tās tīmekļa lietotājiem. PaaS pakalpojumi ir mitināti mākonī un lietotāji tiem var piekļūt ar tīmekļa pārlūka palīdzību

**RDS (Relational Database Service)** – *AWS* relāciju datu bāzu serviss, kas paredzēts datu bāžu uzstādīšanai, darbībai un mērogošanai mākonī

**REST (Representational state transfer)** – tīmekļa arhitektūras pamatprincips, kas nodrošina savietojamību starp datorsistēmām tīmeklī

**S3 (Simple Storage Service)** – *AWS* piedāvātais pakalpojums, kas nodrošina klientiem datu glabātuvī ar vienkāršu tīmekļa saskarni, lai glabātu un izgūtu datus tīmekļa vidē

**SSD** – cietvielu disks; datu krātuves risinājums, kurš izmanto integrētu shēmu montāžu kā atmiņu, pastāvīgai datu glabāšanai

**SaaS (Software as a Service)** – programmatūras sadales metode, kurā trešās puses pakalpojumu sniedzējs lietotni izmitina un kurā programmatūra ir pieejam klientiem internetā

**TCP (Transmission control protocol)** – tīmekļa komunikācijas protokols, kurš paredzēts datu pakešu sūtīšanai internetā

**UDP (User Datagram Protocol)** – daļa no TCP/IP protokolu kopas, kas tiek izmantota datu pārvietošanā

**URI** – vienotais resursu identifikators

**EBS (Elastic Block Store)** – *Amazon Web Services* mākoņglabāšanas pakalpojums

**HDD** – datu glabāšanas ierīce, kas paredzēta pastāvīgai datu glabāšanai datorā

## IEVADS

Mākoņskaitļošanas pakalpojumu straujā attīstība ir radījusi virkni juridisku problēmu. Šīs problēmas galvenokārt balstās uz datu pārvaldību un kontroli - rodas jautājumi par to, kam un kādos gadījumos potenciāli ir piekļuve datiem mākonī. Nesenā pagātnē dažādu valstu likumdevēju vidū radušies priekšlikumi par reģionāliem mākoņiem – mākoņskaitļošanas pakalpojumiem, kas paredzēti tikai noteiktam ģeogrāfiskam apgabalam. Viens no piemēriem ir Eiropas mākonis [HON 16].

Minētie jautājumi būtiski ietekmē tīmeklī bāzētu globālu sistēmu izstrādātājus, jo, strādājot ar personu datiem, kuri nāk no dažādiem pasaules reģioniem, jāņem vērā atbilstošā reģiona likumdošana, kā arī klientu vēlmes par to, ko ar datiem ļauts darīt. Lai izstrādājamo sistēmu būtu iespējams pielāgot šiem nosacījumiem, ir jāpēta plānoto virtuālo mašīnu atrašanās vietu valstu likumdošana saistībā ar datu glabāšanu un apstrādi. Papildus šiem uzdevumiem, sistēmas plānošanas gaitā jāveic padziļināta potenciālās sistēmas arhitektūras izpēte, nepieciešams iedziļināties trešo pušu sniegtajos pakalpojumos, ko plānots izmantot sistēmas izstrādē un uzturēšanā. Liela nozīme sistēmas izstrādes efektivitātē ir arī izvēlēta mākoņskaitļošanas pakalpojumu sniedzēja izvēle, jo katram no tiem ir savas priekšrocības un trūkumi attiecībā uz multireģionālu sistēmu uzturēšanu.

Darbā pētāmā problēma ir, kā veiksmīgi sniegt informāciju tehnoloģiju pakalpojumus klientiem, kuri atrodas dažādās pasaules valstīs un kuru likumdošana nosaka atšķirīgas prasības pret personu datu glabāšanu un apstrādi. Šī darba mērķis ir izpētīt mākoņskaitļošanas pakalpojumu sniedzēju piedāvātās iespējas, kuras ietekmē multireģionālas sistēmas, kā arī izpētīt problēmas, kas rodas multireģionālas, uz tīmekli bāzētas sistēmas izstrādes procesā.

Darba uzdevumi ir:

- 1) izpētīt, kas ir mākoņskaitļošana un kādas ir mākoņskaitļošanas izmantošanas priekšrocības, salīdzinot ar personīgas infrastruktūras izveidi un uzturēšanu;
- 2) izpētīt juridiskās problēmas saistībā ar multireģionālas sistēmas uzturēšanu mākonī;
- 3) salīdzināt 3 populāru mākoņskaitļošanas pakalpojumu sniedzēju piedāvājumus, balstoties uz multireģionālas sistēmas vajadzībām;
- 4) veikt gadījuma izpēti uzņēmumam, kas veicis migrāciju uz multireģionāla tipa sistēmu un turpina tās izstrādi.

Šis pētījums sastāv no ievada, saturiskās daļas, rezultātiem, secinājumiem, kā arī darba beigās ir pieejams izmantotās literatūras saraksts. Darba saturiskā daļa ir sadalīta 4 nodaļās:

- 1) pirmajā nodaļā tiek apskatīta mākoņskaitļošanas būtība un tās priekšrocības;

- 2) otrajā nodaļā tiek dziļāk apskatītas juridiskās problēmas, ar kurām nākas sastapties, strādājot ar multireģionālu sistēmu;
- 3) trešajā nodaļā tiek veikts 3 populāru mākoņskaitļošanas pakalpojumu sniedzēju salīdzinājums;
- 4) ceturtajā nodaļā ir aprakstīta gadījuma izpēte ziedojumu piesaistes platformai *Funderful*.

# 1. MĀKOŅSKAITĻOŠANA

Nodaļā tiek aplūkots mākoņskaitļošanas jēdziens un īsa notikumu vēsture, kas rosināja šīs tehnoloģijas attīstību. Turpinājumā tiek aprakstīti mākoņskaitļošanas pakalpojumu līmeņi.

## 1.1. Mākoņskaitļošanas jēdziens un vēsture

Pakāpeniski attīstoties tīklošanas, joslas platumā, resursu pārvaldības un virtualizācijas tehnoloģijām, radusies jauna IT industrijas nozare - mākoņskaitļošana. Mākoņskaitļošana ir skaitļošanas veids, kurā datu glabāšanai, pārvaldīšanai un apstrādei lokālu serveru vietā tiek izmantoti internetā izvietoti attālināti serveri.

*1.1. tabula*

### Mākoņskaitļošanas attīstības vēsture

1961. gads	Džons Makartnijs apsver iespēju, ka skaitļošana nākotnē varētu būt publiski pieejama.
1967. gads	<i>IBM</i> atklāj programmatūru “CP-67”; viens no <i>IBM</i> pirmajiem lieldatoru operētājsistēmu virtualizācijas mēģinājumiem.
1999. gads	<i>Salesforce.com</i> iepazīstina ar ideju izmantot tīmekli biznesa līmeņa lietotnēm.
1999. gads	Karla Keselmana un Īana Fostera darba rezultātā tiek ieviests termins “režģiskā skaitļošana”.
2002. gads	<i>AWS</i> piedāvā vairākus mākonī pieejamus pakalpojumus, tai skaitā datu uzglabāšanu un skaitļošanu.
2006. gads	<i>AWS</i> plašākai publikai piedāvā komerciālu tīmekļa servisu “EC2”, kuru savu lietotņu darbināšanai izmanto mazi uzņēmumi un privātpersonas.
2008. gads	Tiek popularizēti privātu mākoņu modeļi.
2009. gads	Mākoņskaitļošanas pakalpojumu sniedzēji piedāvā uz tīmekļa pārlūkiem bāzētas biznesa līmeņa lietotnes.

Mākoņskaitļošanas straujā attīstība [HAR 14] izskaidrojama ar nepārtraukti augošo pieprasījumu pēc ātrākas un lētākas skaitļošanas. Mākoņskaitļošanas pakalpojumu lietotājiem tehnoloģija sniegtu tādas priekšrocības kā iespēju ātri pielāgot infrastruktūru sistēmas

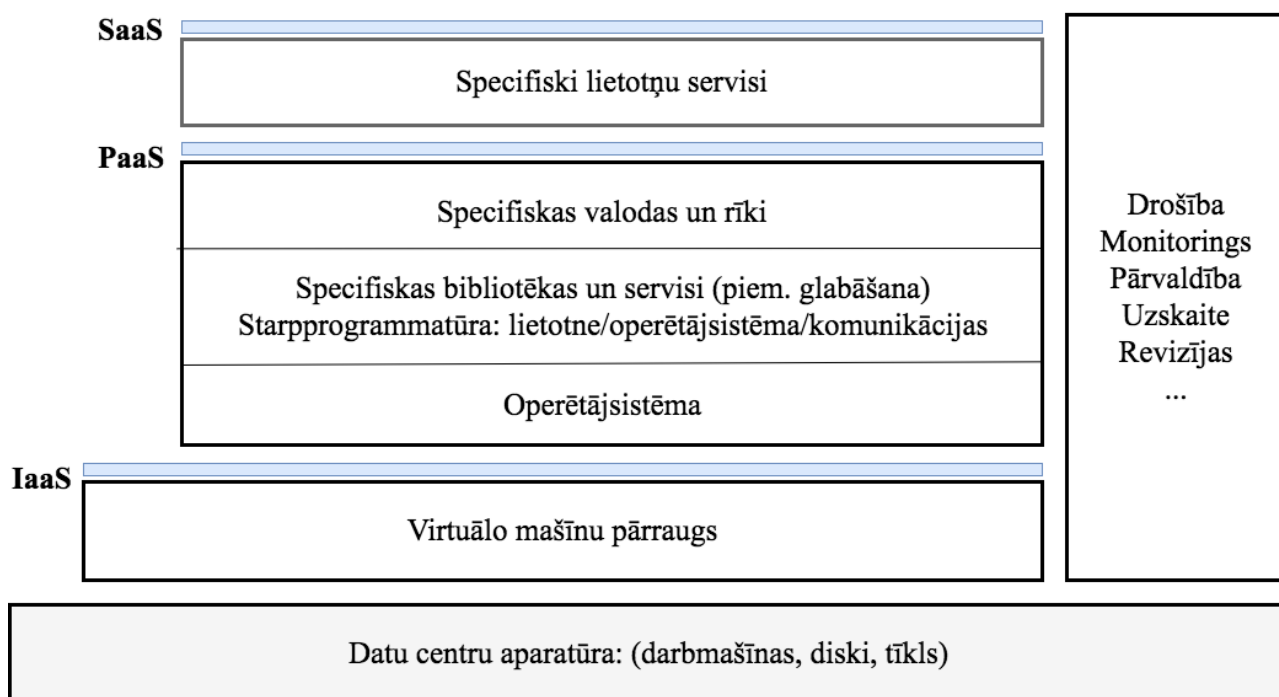
noslogojumam, samazinātas izmaksas, kā arī plašākas iespējas eksperimentēt ar jauniem servisiem. Kā redzams tabulā 1.1, ar mākoņskaitļošanu saistītām koncepcijām tika eksperimentēts jau 1967. gadā, taču straujākā attīstība novērojama ar 1999. gadu. Trīs gadus vēlāk, 2002. gadā, *Amazon* jau spēja piedāvāt risinājumu, kurš plašāk pieejams publikai kļuva 2006. gadā.

Mākoņskaitļošana ietver mākoņa nodrošinātāju, kurš piedāvā pakalpojumu, tehnisko resursu nodrošinājumu un pārvaldību nomnieku starpā. Nomnieki ir pakalpojuma lietotāji, kas patērē mākoņa resursus caur tiešām saistībām ar pakalpojuma sniedzēju. Pakalpojuma sniedzēja biznesa modelis ļauj sniedzējam piedāvāt servisu par zemāku cenu, jo resursi tiek dalīti nomnieku starpā. Turklāt nomnieki spēj ietaupīt finansiālos resursus, maksājot tikai par resursiem, kuri tiem ir nepieciešami, tādejādi likvidējot augstās uzturēšanas pamatizmaksas un iegūstot lielāku elastību ar iespēju ātri paplašināt vai samazināt izmantotos skaitļošanas resursus, balstoties uz pieprasījuma svārstībām.

Sistēmas gala lietotāji var mijiedarboties ar mākoņskaitļošanas pakalpojuma sniedzēju tieši vai netieši. Pakalpojumu sniedzēji var arī izmantot citu pakalpojumu sniedzēju servisu kā apakšpakalpojumus ar mērķi nodrošināt savus piedāvātos servisu, līdz ar to skaitļošanas pakalpojumu sniegšanas procesā ir vienlaicīgi iesaistītas vairākas ieinteresētās puses, kas būtiski sarežģī pakalpojuma izmantošanas juridisko pusi. Papildus tam ir iespējams, ka dažādas valstis pretendē uz tiesībām regulēt visus vai daļu no šiem pakalpojumiem.

## **1.2. Mākoņskaitļošanas pakalpojumu līmeņi**

Mākoņskaitļošanas pakalpojumu sniedzēji savus pakalpojumus piedāvā, balstoties uz dažādiem modeļiem, no kuriem 3 standarta modeļi ir “Infrastruktūra kā serviss” (IaaS), “Platforma kā serviss” (PaaS) un “Programmatūra kā serviss” (SaaS) [MEL 11]. Šie modeļi atšķiras ar abstrakcijas līmeņiem jeb kārtām: infrastruktūra, platforma un programmatūra, taču tās var arī nebūt saistītas. Piemēram, pakalpojumu sniedzējs var nodrošināt klientiem SaaS, kurš implementēts uz fiziskas aparatūras, neizmantojot PaaS un IaaS abstrakcijas līmeņus. Turpretī iespējams darbināt programmu IaaS un piekļūt tai tieši, neizmantojot SaaS līmeni.



1.1. att. Mākoņskaitļošanas pakalpojumu līmeņi un ar tiem saistītie piedāvājumi

Attēlā 1.1 vizualizēts mākoņskaitļošanas pakalpojumu sadalījums dažādos līmeņos. Zemākajam abstrakcijas līmenim pieder datu centru aparātūra, augstākajam – lietotņu servisi, kā piemēram, klientu pārvaldības rīks.

### 1.2.1. Infrastruktūra kā serviss

Infrastruktūra kā serviss [BAR 17] ir standartizēts un lielākoties automatizēts serviss, kur pakalpojumu sniedzējs, izmantojot virtuālās mašīnas, klientiem piedāvā skaitļošanas resursus kopā ar datu krātuvēm un tīklošanas iespējām.

Mākoņskaitļošanā virtuālā mašīna ir datorsistēmas emulācija. Virtuālās mašīnas ir balstītas uz datoru arhitektūrām un nodrošina fiziska datora funkcionalitāti, to implementācija var prasīt specializētu aparāturu un programmatūru.

Nomnieks uz virtuālās mašīnas uzstāda operētājsistēmu (visbiežāk no attēla, kas ietver sevī konfigurētu sistēmu), un ir atbildīgi par savas lietotnes, kā arī citas programmatūras uzstādīšanu un pārvaldīšanu uz uzstādītās operētājsistēmas. Pakalpojuma sniedzējs pārvalda virtuālās mašīnas caur virtuālo mašīnu pārraugu, kurš darbojas nomnieku virtuālajās mašīnās un regulē to mijiedarbību ar pakalpojuma sniedzēja aparāturu. Pakalpojuma sniedzējs spēj izolēt vai kopīgot fiziskos virtuālo mašīnu resursus, kurus nomnieki izmanto. Tas, cik daudz tiek kopīgots, atkarīgs no piedāvājuma, piemēram, “Amazon Dedicated Instances” nodrošina specifiski nomniekam paredzētu aparāturu. Gala lietotāji mijiedarbojas ar pakalpojuma

sniedzēju netieši, caur programmatūru, kuru nodrošina nomnieks. Populārākie IaaS pakalpojumi ir *Amazon EC2*, *Microsoft Azure* un *Google Compute Engine*.

Alternatīva virtuālo mašīnu pārrauga metodei ir konteineri jeb operētājsistēmu virtualizācija [NAG 15] - nomnieki koplieto vienu un to pašu operētājsistēmu, bet to resursu (procesors, atmiņa, ievade/izvade) izmantošana un piešķiršana ir izolētas. Konteineri attiecas gan uz IaaS, gan uz PaaS pakalpojumiem; pakalpojumu sniedzējs piedāvā operētājsistēmu, kura parasti darbojas tieši uz aparatūras kopā ar dažāda veida citiem servisiem. Populāras operētājsistēmu līmeņa virtualizācijas implementācijas ir *LXC*, *HP-UX*, *Parallels Virtuozzo* un *Amazon Docker*.

### 1.2.2. Platforma kā serviss

“Platforma kā serviss” (PaaS) ir pilna izstrādes un izvietojanas vide mākonī. Tā sniedz resursus, ar kuriem iespējams realizēt gan triviālas mākonī bāzētas lietotnes, gan sarežģītas biznesa līmeņa lietotnes. Resursi tiek nomāti no mākoņskaitļošanas pakalpojumu sniedzēja un piekļuve tiem notiek tīmeklī.

Līdzīgi kā IaaS, arī PaaS iekļauj sevī infrastruktūru – serverus, datu krātuves un tīklošanu, taču papildus tam pieejama starpprogrammatūra, izstrādes rīki, biznesa informācijas servisi, datu bāžu pārvaldības sistēmas un citi pakalpojumi un rīki. PaaS serviss ir veidots, lai nodrošinātu pilnu tīmekļa lietotnes dzīvesciklu: izstrādi, testēšanu, izvietojšanu, pārvaldību un atjaunošanu.

PaaS priekšrocības [SGS 16] ir:

- iespēja programmēt augstākā līmenī ar ievērojami mazāku sarežģītību; kopējā lietotnes izstrāde var būt daudz efektīvāka, jo PaaS piedāvā gatavu infrastruktūru;
- iespēja mērogot pakalpojumu attiecīgi lietotāju pieaugumam vai samazinājumam, izmantojot mehānismus, kurus izstrādājuši industrijas eksperti;
- pakalpojuma izmantošana nereti ļauj izvairīties no dārgu programmatūras licenču pirkšanas un pārvaldīšanas, ja tiek izmantoti rīki, ko piedāvā pakalpojuma sniedzējs.

PaaS potenciālie ierobežojumi [SGS 16]:

- nespēja izmantot pilnas tradicionālo rīku iespējas, piemēram, neierobežotas sasaistes relāciju datu bāžu sistēmu pieprasījumus;
- piesaiste noteiktai platformai.

Populāri PaaS pakalpojumu sniedzēji ir *Microsoft*, *AWS* un *SalesForce*.

### 1.2.3. Programmatūra kā serviss

“Programmatūra kā serviss” pakalpojums nodrošina visu gala lietotāja lietotni, kas nozīmē, ka ne tikai implementācija un izvietošana ir pakalpojuma sniedzēja pārvaldīta, bet pakalpojums iekļauj arī pašu lietotnes funkcionalitāti. Gala lietotāji mijiedarbojas tieši ar pakalpojuma sniedzēja lietotni. Šādam modelim atbilst, piemēram, “Google Mail” vai “Github”. Nomnieki var nomāt servisu, iespējams pat pielāgot tās funkcionalitāti, piemēram, pielāgojot e-veikala maksājumu plūsmu. Vēl kāds ierasts darbības modelis ir, piemēram, universitātēm, kas izmanto tīmekļa e-pasta pakalpojumus, lai nodrošinātu organizāciju ar e-pastu sistēmu, kur programmatūru parasti kontrolē pats pakalpojuma sniedzējs.

Tas, cik izolēti viens no otra ir nomnieki, atkarīgs no piedāvātā pakalpojuma. Iespējams, ka daļai nomnieku pat tiek piešķirta tikai tiem paredzēta infrastruktūra, citos gadījumos visa vai daļa no infrastruktūras var tikt dalīta starp nomniekiem.

## **2. MĀKOŅSKAITĻOŠANAS SERVISU DATU LOKĀCIJAS JURIDISKIE IEROBEŽOJUMI**

Lai arī, izmantojot mākoņskaitļošanas pakalpojumus, dati neglabājas uz personīgās aparatūras, tie tāpat tiek glabāti uz fiziskas iekārtas kādā konkrētā lokācijā, kas nosaka, kādai jurisdikcijai šie dati tiek pakļauti. Ja mākoņskaitļošanas pakalpojumu sniedzējam, kas glabā lietotāja datus, ir serveri ārvalstīs, serverī glabātie dati var tikt pakļauti attiecīgās valsts likumiem. Tā rezultātā vairāki svarīgi ārvalstu likumi var noteikt, kam ir tiesības piekļūt šiem datiem. Nodaļā tiks apskatīti iemesli, kāpēc pakalpojuma sniedzējam var rasties vajadzība glabāt datus dažādās pasaules vietās.

### **2.1. Datu aizsardzības likumi**

Lai ilustrētu, kā tiek piemēroti datu aizsardzības likumi [SUN 14], tiek apskatīts kāds mākoņskaitļošanas serviss, kas nodrošina satura mitināšanas pakalpojumus. Šim servisam ir datu centri visā pasaulē, tostarp arī Lielbritānijā. Ja kāds uzņēmums, kurš ir bāzēts ASV, izmanto šī servisa pakalpojumus, iespējama situācija, ka uzņēmums, kuram nav nekādu darbību ārvalstīs, fiziski glabā datus Lielbritānijā pats to neapzinoties.

“The Data Protection Act 1998” [BIP 98] nosaka personīgo datu aizsardzību informācijai, kas tiek apstrādāta Lielbritānijā. Lai gan likumprojektā noteiktie likumi galvenokārt paredzēti uzņēmumiem, kas darbojas Lielbritānijā, likumprojekta 5(1)(b) sadaļā noteikts, ka likums attiecas arī uz datu turētājiem, kas nav bāzēti Lielbritānijā vai kādā citā Eiropas Ekonomiskās zonas valstī, bet kuri izmanto aparatūru Lielbritānijā datu apstrādei, kas paredzēta kam vairāk kā datu vadīšanai caur Lielbritāniju.

Tas nozīmē, ka, ja ārzemju valsts izmanto aparatūru, kas atrodas Lielbritānijā, lai apstrādātu personas datus, datu apstrādei jāatbilst Lielbritānijas datu aizsardzības likumiem, arī tad, ja uzņēmums nav izveidots un neveic nekādas operācijas Lielbritānijā. Šāds noteikums atrodams arī visu pārējo Eiropas Ekonomiskās zonas, kā arī citās valstīs.

Mākoņskaitļošanas servisu sniedzējam, izvēloties uzstādīt serverus kādā Eiropas Ekonomiskās zonas valstī vai citā valstī ar līdzīgiem datu aizsardzības likumiem, visi dati, kas tiek apstrādāti vai uzglabāti uz šiem serveriem, tiek pakļauti attiecīgās valsts datu aizsardzības likumiem. Šiem likumiem ir plašas prasības, ierobežojumi un aizliegumi attiecībā uz to, ko drīkst un ko nedrīkst darīt ar personas datiem. Mākoņskaitļošanas pakalpojumu sniedzējam var būt nepieciešams reģistrēties valsts datu aizsardzības pārraudzības iestādē, kā arī tam var

tikt aizliegtas kādu noteiktu datu transakcijas. Šo noteikumu neievērošana var rezultēties naudas sodos līdz pat 4% no uzņēmuma apgrozījuma [REP 14].

Datu aizsardzības regulējumu pastiprināšana notikusi arī nesenākā pagātnē Krievijā. No 2015. gada 1. septembra spēkā stājies likums, kas nosaka, ka uzņēmumiem Krievijas pilsoņu dati jāglabā datu bāzēs, kas atrodas Krievijā. Lokalizācijas likuma sfēras apjoms nav precīzi noteikts [DFC 15], bet pakalpojumu sniedzējiem jānodrošina, ka Krievijas pilsoņu datu iegūšanai, sistematizācijai, glabāšanai, atjaunošanai, papildināšanai un izvilksanai jānotiek Krievijas teritorijā esošās datu bāzēs. Likuma nekonkrētums izpaužas gadījumos, kad uzņēmumi iegūst personas datus no Krievijas pilsoņiem, bet tai pat laikā tie neoperē no Krievijas.

Uzņēmumiem, kas strādā ar Krievijas pilsoņu personas datiem, ir arī jāinformē atbildīgā valsts iestāde “Roskomnadzor” par uzņēmuma serveru un datu glabātuvju atrašanās vietām [MED 16]. Domēnu nosaukumi vai interneta adreses, caur kurām iespējams piekļūt Krievijas pilsoņu datiem un kuri neatbilst jaunajiem regulējumiem, var tikt reģistrēti speciālā sarakstā, kā vienības, kas pārkāpj personas datu subjektu tiesības. Turpināšana nepakļauties rezultējas domēna/interneta adreses bloķēšanā.

Jaunie regulējumi ir skāruši tādas populāru tīmekļa vietņu īpašniekus kā *Google*, *Facebook*, *LinkedIn* un *Twitter*. 2016. gada novembrī tīmekļa vietne *LinkedIn* tika bloķēta Krievijā, kur konkrētajā brīdī bija reģistrēti aptuveni 5 miljoni lietotāju. Līdzīgos apstākļos, lai izpildītu Ķīnā noteiktās prasības pret datu drošību, *LinkedIn* izveidoja sistēmu [FRI 14], kas atdalīta no tās, kuru izmanto citās pasaules valstīs.

Līdzīgi datu aizsardzības regulējumi [KUR 15] ir spēkā Ķīnā, Vjetnamā, Indonēzijā un Malaizijā. Arī šajās valstīs pilsoņu personu datiem jāglabājas uz lokāliem serveriem.

## **2.2. Valdības uzraudzība**

Papildus iepriekšējā apakšnodaļā minētajiem datu aizsardzības likumiem, jāapsver arī iespējamība, ka kāda trešā puse vai ārvalstu valdība varētu pieprasīt piekļuvi mākoņskaitļošanas pakalpojuma sniedzēja serverim, kurā glabājas lietotāja dati. Trešās puses, tai skaitā arī valdības, piekļuve datiem ir aizliegta un pat varas iestādēm vai slepenajam dienestam var nebūt pieeja telpām vai aparatūrai bez atbilstošas autorizācijas - pilnvaras vai tiesas rīkojuma formā. Taču ir vietas, kur tā nav [DER 14].

Ja pakalpojuma lietotāju dati tiek glabāti uz serveriem, kas atrodas Indijā, šie serveri ir Indijas likumu pakļautībā. Indijas “Information Technology Act of 2000” [TIT 20] (ar 2009. gada pielikumu) nosaka vairākus aspektus saistībā ar datu aizsardzību un datoru, kā arī tīklu

izmantošanu. 69. nodaļa likumā nosaka, ka valdībai drošības nolūkos, sabiedriskās kārtības kontrolēšanas nolūkos vai kāda pārkāpuma izmeklēšanā, ir ļauts izdot rīkojumus ziņu pārtveršanai, novērošanai un atšifrēšanai no jebkura datora vai komunikāciju ierīces. Apakšnodaļa 69B(1) sniedz valdībai varu autorizēt jebkuru valsts aģentūru novērot un uzkrāt informācijas plūsmu datus, kā arī ģenerētu, pārraidītu, saņemtu un noglabātu informāciju uz jebkura datora. Abos gadījumos nav prasību pēc tiesas pavēles vai citu pilnvaru un šai varai nav ierobežojumu.

Indijas valdībai ir arī iespēja noteikt to kādus datus drīkst un kādus nedrīkst saglabāt. Nodaļa 67C no iepriekšminētā likuma pieprasa uzņēmumiem uzglabāt tādu informāciju, tik ilgā laika posmā un tādā veidā un formātā kā valdība noteikusi.

Lietojot mākoņskaitļošanas pakalpojumus ir iespējams izmantot priekšrocības, ko sniedz valstis ar draudzīgām biznesa vidēm, taču tajā pašā laikā aparatūra un uzglabātie dati tajā var tikt pakļauti attiecīgās valsts valdības novērošanas un uzraudzības likumiem.

Viens no aktuāliem gadījumiem nesenā pagātnē ir *Microsoft* strīds ar ASV valdību [MET 16]. ASV valdība pieprasīja piekļuvi personas e-pastiem, kuri bija saistīti ar izmeklēšanu 2013. gada decembrī. Šie e-pasti tika turēti uz servera Īrijā, nevis ASV. *Microsoft* atteicās izsniegt e-pastus, apgalvojot, ka ASV valdībai nav tiesību pieprasīt šos datus, jo tie tiek glabāti citā valstī un ārpus ASV jurisdikcijas. 2014. gada aprīlī ASV federālais tiesnesis pavēlēja *Microsoft* izsniegt datus un arī pēc šī lēmuma *Microsoft* atteicās sadarboties. 2016. gada 14. aprīlī tiesa sprieda par labu *Microsoft* [POT 15].

Kā pamats informācijas pieprasīšanai bija “1986 Stored Communications Act (SCA)”, kas dod tiesības valdībai likt uzņēmumiem izsniegt jebkādus tiem pieejamus datus, ja tie var palīdzēt ar lietu izmeklēšanu. *Microsoft* ir jau iepriekš ticis vainots necienīgā izturēšanās pret tiesu, apgalvojot, ka uzņēmumam nav jāizsniedz prasītā informācija, jo dati nepieder uzņēmumam, bet tiek tikai uzturēti uz *Microsoft* aparatūras. Aplūkojot problēmu no biznesa puses, ja *Microsoft* brīvi izdotu datus ASV valdībai, tas liktu klientiem domāt, ka viņu dati nav pilnībā privāti. Tā rezultātā klienti izmantotu tādu uzņēmumu pakalpojumus, kuri operē valstīs, kurām nav datu izdošanas vienošanās ar ASV.

### **2.3. Jurisdikcijas nozīme mākoņskaitļošanā**

Pastāv uzskats, ka jautājumi, kuri skar mākoņskaitļošanas pārvaldību, būtu jāadresē balstoties uz to, kam ir pieeja datiem un kurām valstīm ir jurisdikcija pār šīm personām, nevis pieņemot pakalpojuma sniedzēja atrašanās vietu kā galveno aspektu. Juridiskie konflikti, kuri saistīti ar mākoņskaitļošanu un internetu, jau ilgi bijusi problēma tehnoloģiju pasaulē [WOT

15]. Viena no problēmām ir tas, ka nomnieks, pakalpojuma sniedzējs un citas iesaistītās puses, atkarībā no sniegtā pakalpojuma, var tikt pakļautas dažādām jurisdikcijām. Valdības aizvien vairāk mēģina piemērot savus likumus ārpus savas valsts teritorijām, taču šo problēmu nevar atrisināt ar tehnoloģijām un ir nepieciešama starptautiska politiska vienošanās [RIC 15].

Vairāki datu pārvaldības politikas aspekti ir neatkarīgi no tehniskajām interesēm, piemēram, daļa juridisko, ekonomisko un sociālo aspektu, taču tāpat ir daļa aspektu, kuri tiešā veidā ietekmē tehnisko problēmas pusi. Viens no likumsakarīgiem variantiem ir mākoņskaitļošanas regulēšanā ņemt vērā divas lietas:

- pakalpojuma sniedzēja spēja piekļūt lietotāju datiem;
- jurisdikcija, kurā pakalpojuma sniedzējs operē.

Pirmais no minētajiem punktiem attiecas uz pakalpojuma sniedzēja pieeju saprotamiem lietotāju datiem, piemēram, datu glabāšanas serviss, kas glabā šifrētus datus, bet kuram nav pieejas atslēgām, ir uzticamāks par tādu, kam ir piekļuve atšifrēšanas atslēgām, vai tādu kurš operē ar nešifrētiem datiem. Šis punkts var tikt risināts ar tehniskiem līdzekļiem.

Attiecībā uz otro punktu, jurisdikcijai nav jābūt attiecināmai uz fizisku lokāciju. Potenciāli pakalpojumu sniedzēji varētu piedāvāt pakalpojumus, izpildot nosacījumus vai nu kādā noteiktā jurisdikcijā, kas nav atkarīga no vietas vai arī pakalpojumu sniedzēju jurisdikcijā. Papildus būtu jāņem vērā arī pakalpojumu sniedzēja korporācijas atrašanās vietas jurisdikcija. Arī šīs datu pārvaldības problēmas ir risināmas ar tehniskiem līdzekļiem, piemēram, ļaujot informācijai pārvietoties tikai noteiktā “virtuālā jurisdikcijā”, pieņemot, ka ierobežojumus ir iespējams nedefinēt.

### **3. MĀKOŅSKAITĻOŠANAS PAKALPOJUMU SNIEDZĒJU SALĪDZINĀJUMS MULTIREĢIONĀLĀS INFORMĀCIJAS SISTĒMĀS**

Balstoties uz datiem no “IDC” pētījuma [WPC 17], kopējās mākoņskaitļošanas servisu un infrastruktūras izmaksas 2017. gadā sasniegs 122,5 miljardus dolāru, kas no 2016. gada ir 24,5% palielinājums. Pēdējos gados, mākoņskaitļošanas popularitātei augot, arī mākoņskaitļošanas pakalpojumu sniedzēju skaits ir ievērojami palielinājies. Izvēlēties sev piemērotāko pakalpojumu sniedzēju ir īpaši svarīgi, jo atkarībā no sistēmas sarežģītības, migrācija no viena pakalpojumu sniedzēja uz citu var būt sarežģīta un laikietilpīga. Nodaļā tiks apskatīti dažādi faktori, pēc kuriem izvēlēties sev piemērotāko mākoņskaitļošanas pakalpojumu sniedzēju multireģionālas sistēmas izstrādei, kā arī tiks veikts trīs populāru mākoņskaitļošanas pakalpojumu sniedzēju salīdzinājums. Kā viens no salīdzināšanas kritērijiem atsevišķos aspektos papildus tiks izmantots tīmekļa vietnes *ShareStack* [STA 17] komūnas vērtējums.

Šī pētījuma ietvaros par multireģionālām sistēmām tiek sauktas tādas sistēmas, kurām juridisku, tehnisku, finansiālu vai citu iemeslu dēļ dati tiek glabāti un apstrādāti dažādos pasaules reģionos, kuri pakļaujas attiecīgo reģionu jurisdikcijām.

#### **3.1. Mākoņskaitļošanas pakalpojumu sniedzēji**

Apakšnodaļā tiks aprakstīti trīs mākoņskaitļošanas pakalpojumu sniedzēji, kuru salīdzinājums tiek veikts pētījuma turpinājumā.

##### **3.1.1. Amazon Web Services**

*Amazon Web Services* (turpmāk tekstā - *AWS*) ir *Amazon.com* meitasuzņēmums, kas piedāvā mākoņskaitļošanas pakalpojumus. Šie pakalpojumi pieejami no 16 ģeogrāfiskiem reģioniem visā pasaulē. *AWS* pakalpojumos ietilpst “Elastic Compute Cloud”, pazīstams arī ar nosaukumu “EC2”, kā arī “Amazon Simple Storage Service” pazīstams arī kā “S3”. Kopš 2016. gada *AWS* piedāvā vairāk nekā 70 plaša spektra pakalpojumus [CPR 17], tai skaitā skaitļošanas, glabāšanas, tīklošanas, datubāžu, analītiskus, izvietojšanas, pārvaldības, kā arī izstrādātāju rīkus un rīkus priekš lietu interneta. *AWS* tirgū savus pakalpojumus pasniedz kā veidu, kā nodrošināt lielu skaitļošanas kapacitāti ātrāk un lētāk par fiziskas serveru fermas izveidi. *AWS* kļuva publiski pieejams 2006. martā.

### 3.1.2. Microsoft Azure

*Microsoft Azure* [WIA 17] ir *Microsoft* piedāvātais mākoņskaitļošanas serviss, kurš paredzēts lietotņu un servisu izstrādei, izvietošanai un pārvaldībai, izmantojot *Microsoft* pārvaldītu datu centru tīklu. Tas nodrošina SaaS, PaaS un IaaS darbību un atbalsta vairākas programmēšanas valodas, rīkus un satvarus, tai skaitā *Microsoft*, kā arī trešo pušu programmatūru un sistēmas. *Azure* tika izziņots 2008. gada oktobrī un kļuva pieejams 2010. gada 1. februārī kā *Windows Azure*, taču 2014. gada 25. martā tika pārsaukts par *Microsoft Azure*.

### 3.1.3. Google Cloud Platform

*Google Cloud Platform* [ATG 17] ir *Google* pakalpojums, kas piedāvā mitināšanu uz tādas pašas infrastruktūras, kādu *Google* izmanto iekšējiem produktiem kā *Google Search* un *YouTube*. *Google Cloud Platform* ir daļa no *Google Cloud* biznesa līmeņa servisiem un piedāvā vairākus modulārus mākonī izvietotus servissus ar dažādiem izstrādes rīkiem, piemēram, izvietošanu un skaitļošanu, datu glabāšanu, kā arī tulkošanas un prognozēšanas programmsaskarnes. Galvenie *Google Cloud Platform* pakalpojumi ir *Google Compute Enging* (IaaS), *Google App Engine* (PaaS), *Google Cloud Storage* - mākoņkrātuve, kas paredzēta lielām nestrukturētām datu kopām, kā arī *Google Container Engine – Docker* konteineru pārvaldības rīks. *Google Cloud Platform* publiski kļuva pieejams 2011. gada 6. oktobrī.

## 3.2. Mākoņskaitļošanas pakalpojumu sniedzēju salīdzinājuma kopsavilkums

Mākoņskaitļošanas pakalpojumu sniedzēju salīdzināšanai tiek izmantoti 10 faktori: izmaksu apmērs, automatiskās izvietošanas iespējas, automatiskās mērogojamības iespējas, monitoringa rīki, slodzes līdzsvarošanas iespējas, mākoņkrātuves, komūnas atbalsts, tehniskais atbalsts, datu centru lokācijas un domēnu nosaukumu pakalpojumu iespējas.

## Mākoņskaitļošanas pakalpojumu sniedzēju salīdzinājuma kopsavilkums

Faktors	AWS	Google Cloud Platform	Microsoft Azure
Izmaksas	Zemākā cena - 3 gadījumi Augstākā cena - 1 gadījums	Zemākā cena - 3 gadījumi Augstākā cena - 7 gadījumi	Zemākā cena - 6 gadījumi Augstākā cena - 4 gadījumi
Automātiskās izvietojšanas iespējas	<i>AWS CodeDeploy</i> a) GitHub integrācija b) 2 veidu izvietojšanas - <i>inplace</i> un <i>blue/green</i> c) neatkarīgs no platformas d) <i>Stackshare</i> balsis: 28	<i>Deployment manager</i> a) izvieta uz vairākiem resursiem paralēli b) izmaiņu priekšskatījums c) izmantojams ar <i>Google Cloud Storage</i> , <i>Google Compute Engine</i> un <i>Google Cloud SQL</i> d) <i>Stackshare</i> balsis: nav	a) 3 dažādi izvietojšanas procesi b) nav atsevišķa rīka automātiskai izvietojšanai c) <i>Stackshare</i> balsis: nav
Automātiskas mērogojamības iespējas	a) instanču veidošana ar zonu ierobežojumiem b) paredzamas noslodzes grafiks c) <i>Amazon CloudWatch</i> trigeri d) vertikāla un horizontāla mērogošana	a) nav iespējams izmantot vairākus nosacījumus reizē b) <i>Google Cloud Pub/Sub</i> integrācija c) pieejama tikai horizontāla mērogošana	a) iespēja uzstādīt sarežģītus, pārklājošus nosacījumus b) e-pastu notifikācijas c) <i>webhook</i> integrācijas d) vertikāla un horizontāla mērogošana
Monitoringa rīki	<i>Amazon CloudWatch</i> a) neprasa nekādas papildus darbības, lai uzstādītu b) automātiskas mērogošanas integrācijas	<i>Google Stackdriver</i> a) iespējama integrācija <i>AWS</i> un citās atvērtā pirmkoda pakotnēs kā <i>MongoDB</i> , <i>Apache</i> , <i>Nginx</i> , <i>Elasticsearch</i> . b) iespējams apvienot metrikas no vairākiem pakalpojumu sniedzējiem	<i>Azure Monitor</i> a) datu vizualizācijai datus var novirzīt uz <i>Azure Application Insights</i> , <i>Microsoft PowerBI</i> vai kādu citu trešās puses datu vizualizācijas rīku, izmantojot datu straumēšanu vai nolasišanu no arhīva <i>Azure</i> noliktavā.
Slodzes līdzsvarošana	<i>Elastic Load Balancing</i> a) 2 slodzes līdzsvarošanas iespējas - primitīva un avancēta b) iespēja novirzīt trafiku, balstoties uz lietotnes līmeņa informāciju	<i>Google Cloud Load Balancer</i> a) iespēja līdzsvarot HTTP(S), SSL un TCP/UDP trafiku b) reģionāla un globāla slodzes līdzsvarošana	<i>Azure Load Balancer</i> a) dalītā slodzes līdzsvarošana b) dinamiski pielāgojas infrastruktūras mērogošanai uz augšu vai uz leju

Faktors	AWS	Google Cloud Platform	Microsoft Azure
	c) transporta slāņa un lietotnes slāņa slodzes līdzsvarošana d) virtuālās mašīnas piesaistīšana lietotājam, izmantojot sīkfailus	c) slodzes līdzsvarošanas žurnālfaili un iespēja tos eksportēt uz citiem <i>Google</i> servisiem	
Mākoņkrātuves	<i>Simple Storage Solution, Elastic Block Storage</i> a) S3 integrācija ar citiem AWS servisiem b) S3 pieejamība – 99,99%, uzticamība 99,999999999%, c) EBS pieejamība 99,999%. d) EBS automātiska sējumu replikācija attiecīgajā reģionā d) <i>Stackshare</i> balsu skaits: 78 + 1630	<i>Google Cloud Storage</i> a) paziņojumi ar aizķerēm b) nodrošināta 99.95% pieejamība multireģionālai glabāšanai c) <i>Stackshare</i> balsu skaits: 46	<i>Azure Storage</i> a) aukstā un siltā glabātuve b) izmaksas veidojas tikai no izmantotā atmiņas daudzuma c) unikāla kešdarbības arhitektūra d) <i>Stackshare</i> balsu skaits: 30
Komūnas atbalsts	<u><i>StackOverflow</i></u> Jautājumu skaits: 83 Jautājumu balsu skaits: 284 Atbilžu skaits: 39 Akceptēto atbilžu skaits: 19  <u><i>Google</i></u> Aptuvenais atgriezto rezultātu skaits: a) 330 000 b) 164 000 c) 169 000	<u><i>StackOverflow</i></u> Jautājumu skaits: 8 Jautājumu balsu skaits: 7 Atbilžu skaits: 3 Akceptēto atbilžu skaits: 3  <u><i>Google</i></u> Aptuvenais atgriezto rezultātu skaits: a) 2 470 000 b) 563 000 c) 897 000	<u><i>StackOverflow</i></u> Jautājumu skaits: 21 Jautājumu balsu skaits: 18 Atbilžu skaits: 9 Akceptēto atbilžu skaits: 7  <u><i>Google</i></u> Aptuvenais atgriezto rezultātu skaits: a) 371 000 b) 159 000 c) 324 000
Tehniskais atbalsts	Bezmaksas piedāvājums: a) pieeja 4 galvenajām <i>AWS Trusted Advisor</i> programmas daļām. Pamata maksas piedāvājums: b) cena: \$29/mēnesī,	Bezmaksas piedāvājums: a) tehniskais atbalsts bezmaksas izmēģinājuma laikā. Pamata maksas piedāvājums: a) cena: \$150/mēnesī,	Bezmaksas piedāvājums: a) iespēja sazināties ar <i>Azure</i> mikroblogošanas vietnē <i>Twitter</i> , b) tehniskais atbalsts <i>Azure</i> pakalpojumu problēmām, izmantojot rīku <i>Resource</i>

<b>Faktors</b>	<b>AWS</b>	<b>Google Cloud Platform</b>	<b>Microsoft Azure</b>
	c) sākotnējās atbildes laiks: līdz 12 stundām.	b) sākotnējās atbildes laiks: līdz 4 stundām.	<i>Health</i> . Pamata maksas piedāvājums: c) cena: \$29/mēnesī, d) sākotnējās atbildes laiks: līdz 8 stundām.
Pašreizējās datu centru lokācijas (iekavās norādīts lokāciju daudzums ieskaitot nākotnē plānotās)	ASV rietumu krasts - 3 ASV centrālā daļa - 0 ASV austrumu krasts - 3 Dienvidamerika - 1 Eiropa - 3 (5) Āzija - 5 (6) Austrālija - 1 Kanāda - 0 <b>Kopā - 16 (27)</b>	ASV rietumu krasts - 1 (1) ASV centrālā daļa - 1 ASV austrumu krasts - 1 (2) Dienvidamerika - 0 (1) Eiropa - 1 (5) Āzija - 3 (4) Austrālija - 0 (1) Kanāda - 0 <b>Kopā - 7 (21)</b>	ASV rietumu krasts - 2 (1) ASV centrālā daļa - 5 (6) ASV austrumu krasts - 2 Dienvidamerika - 1 Eiropa - 6 (8) Āzija - 11 Austrālija - 2 Kanāda - 2 <b>Kopā - 31 (46)</b>
Domēnu nosaukumu pakalpojumi	<i>Route 53</i> a) programmsaskarne b) DNS slodzes līdzsvarošana ar pielāgojamiem svāriem c) <i>Stackshare</i> balsu skaits: 553	<i>Cloud DNS</i> a) REST programmsaskarne b) <i>Stackshare</i> balsu skaits: 30	<i>Azure DNS</i> a) nav programmsaskarnes b) <i>Stackshare</i> balsu skaits: nav

Mākoņskaitļošanas pakalpojumu sniedzēju salīdzinājuma kopsavilkums attēlots tabulā 3.1. Tabulā katram faktoram atzīmētas pakalpojuma sniedzēja piedāvājuma priekšrocības, vietām tiek atzīmēti nopietni trūkumi, kā arī eksperimentu rezultāti, piemēram, komūnas atbalsta faktora gadījumā. Turpmākajās apakšnodaļās katrs no faktoriem tiek iztirzāts plašākā apmērā.

### 3.3. Datu centru lokācijas

Multireģionālu sistēmu uzturēšanā svarīgi ir apzināt mākoņskaitļošanas pakalpojuma sniedzēja pieejamās datu centru lokācijas, jo no tā var būt atkarīgs, vai nomnieks varēs sniegt savus pakalpojumus klientiem, nepārkāpjot klienta valsts vai reģiona likumus. Šajā aspektā

*Microsoft Azure* ir izteikts līderis, kuram pieejami datu centri visos galvenajos reģionos, tai skaitā arī Kanādā un Dienvidamerikā [AZR 17].

Kaut arī *AWS* ir vecākais no apskatītajiem mākoņskaitļošanas pakalpojumu sniedzējiem, tam ir gandrīz uz pusi mazāk datu centru [AGI 17]. Tie atrodas stratēģiski līdzīgās pozīcijās kā *Microsoft Azure* piedāvājums. Piemēram, abiem vienīgie datu centri Dienvidamerikā atrodas Sanpaulu, Brazīlijā, kur arī *Google* nākotnē plāno paplašināties.

Visšaurākais piedāvājums šobrīd pieejams no *Google* [CLL 17], kurš mākoņskaitļošanas tirgū ienācis kā pēdējais no apskatītajiem. Tam pieejami tikai 7 datu centri, tuvākā nākotnē plānoti vēl 14. *Google Cloud* nav pieejami tādi reģioni, kā Dienvidamerika un Austrālija, taču ar salīdzinoši maz datu centriem ir nosepts Āzijas austrumu krasts, ASV, kā arī ir pieejams datu centrs Eiropas vidusdaļā.

Svarīgi piebilst, ka nevienam no apskatītajiem pakalpojumu sniedzējiem nav pieejami un nav plānoti datu centri Āfrikā, kur tirgus pieprasījumu aizpilda citi pakalpojumu sniedzēji, kā, piemēram, *IBM*, kurš 2016. gadā [IBM 16] atvēra datu centru Johannesburgā, Dienvidāfrikā.

### 3.4. Komūnas atbalsts

Komūnas atbalsts ir nozīmīgs faktors, ko izvērtēt pirms sākt lietot jaunu tehnoloģiju, programmatūru vai pakalpojumu, jo programmatūras izstrādē nereti izstrādātāji sastopas ar problēmām, kuras patstāvīgi atrisināt ir grūti.

Pakalpojuma sniedzēja komūnas atbalsta noteikšanai tiek izmantoti divi rādītāji:

- 1) tīmekļa vietnes *StackOverflow* jautājumu un atbilžu skaits kā meklējamo frāzi izmantojot “[pakalpojuma sniedzēja nosaukums] multi region”,
- 2) *Google* meklētāja atgriezto rezultātu skaits kā meklējamās frāzes izmantojot:
  - “[pakalpojuma sniedzēja nosaukums] multi region architecture”
  - “[pakalpojuma sniedzēja nosaukums] multi region deployment”
  - “[pakalpojuma sniedzēja nosaukums] multi region setup”

*StackOverflow* tiek izmantota saīsināta frāze, jo ne visiem aplūkotajiem pakalpojumu sniedzējiem tika atgriezti rezultāti.

Rezultāti (skat. tabulu 3.1.) ievērojami atšķiras, ja tiek salīdzināti *Google* un *StackOverflow* rādītāji. Visaktīvāk uzdotie jautājumi *StackOverflow* ir saistībā ar *AWS*, taču vismazāk jautājumu, atbilžu un balsu ir jautājumiem saistībā ar *Google Cloud*. Aplūkojot *Google* meklētāja rezultātus, uz visiem vaicājumiem visvairāk atbildes tika atgrieztas saistībā

ar *Google Cloud*, kas visdrīzāk gan ir saistīts ar to, ka *Google* piedāvā abus šos pakalpojumus. *Azure* un *AWS* rezultāti ir ļoti līdzīgi ar nelielu *Azure* pārsvaru.

Balstoties uz apkopotajiem rezultātiem, var secināt, ka *AWS* ir lielāks komūnas atbalsts salīdzinājumā ar *Microsoft Azure* un *Google Cloud*.

### 3.5. Monitoringa rīki

Darbā ar multireģionālu sistēmu nākas uzturēt daudz instanču dažādās pasaules vietās, kuru darbības nodrošināšanai var palīdzēt labi izstrādāti monitoringa rīki. Ar tiem iespējams prognozēt problēmsituācijas un laicīgi tās neitralizēt. Visi trīs apskatāmie mākoņskaitļošanas pakalpojumu sniedzēji piedāvā sevis izstrādātus monitoringa rīkus, taču ir pieejami arī trešo pušu risinājumi.

*AWS* piedāvātais rīks saucas *CloudWatch* [ACW 17]. *CloudWatch* var tikt izmantots, lai sekotu līdzī daudziem servera statusa rādītājiem, ievāktu un monitorētu žurnālus, uzstādītu trauksmes signālus un automātiski reaģētu uz izmaiņām klienta *AWS* resursos. Ar *CloudWatch* iespējams monitorēt tādas *AWS* resursus kā *Amazon EC2* virtuālās mašīnas, *Amazon DynamoDB* tabulas, *Amazon RDS* datu bāzu instances, kā arī individuāli veidotas metrikas un žurnālus, kurus ģenerē nomnieka lietotnes vai servisi. *CloudWatch* var tikt izmantots, lai iegūtu detalizētu pārskatu pār resursu patēriņu, lietotņu darbību un sistēmas infrastruktūras statusu.

*Microsoft Azure* piedāvātais rīks saucas *Azure Monitor* [OOM 17]. *Azure Monitor* ļauj ievākt veiktspējas un utilizācijas datus, aktivitātes un diagnostikas žurnālus, kā arī sniedz iespēju saņemt notifikācijas no *Azure* resursiem. *Azure* portālā ir iespējams apskatīt un analizēt monitoringa datus un uzstādīt automatizētas darbības, kas balstītas uz trigeriem un trauksmes signāliem. *Azure Monitor* ir integrācijas iespējas ar citiem analītikas un monitoringa rīkiem kā *OMS Insight & Analytics* un citiem.

*Google* piedāvātais monitoringa rīks saucas *Stackdriver* [STM 17]. *Stackdriver* nodrošina monitoringa, reģistrēšanas un diagnostikas funkcijas. Tas sniedz ieskatus uz mākoņskaitļošanas servisa balstītu lietotņu darbībā un ļauj ātrāk atrast un atrisināt problēmas. *Stackdrive* ir iespējams integrēt ne tikai *Google* platformā, bet arī *AWS* un populārās atvērtā pirmkoda pakotnēs. Ar šādu pozīciju *Google* atbalsta aizvien augošu daļu uzņēmumu, kas izvēlas savām sistēmām līdzās izmantot vairākus mākoņskaitļošanas pakalpojumu sniedzējus. *Stackdrive* pieejami spraudņi tādiem atvērtā pirmkoda serveriem kā *MongoDB*, *Apache*, *Nginx*, *Elasticsearch*. Funkcionalitāte, kas *Stackdrive* izceļ salīdzinājumā ar pārējiem diviem rīkiem, ir iespēja eksportēt žurnālus uz ārējiem servisiem, kā, piemēram, *Google Cloud*

*Storage*, *Google BigQuery* datu kopu, *Google Cloud Pub/Sub* vai jebkādu trīs iepriekš minēto kombināciju. Žurnāli, kas eksportēti uz *BigQuery* var tikt pārmeklēti un analizēti. *Pub/Sub* var tikt izmantots, lai eksportētu žurnālfailus uz trešo pušu servisiem vai REST galapunktiem.

Visi no apskatītajiem monitoringa rīkiem ir līdzīgi, taču katram ir savas stiprās puses, kuras katram pakalpojumu lietotājam atkarībā no situācijas un sistēmas specifika jāizvērtē pašam.

### 3.6. Domēnu nosaukumu pakalpojumi

Darbā ar multireģionālu sistēmu var ikdienā nākties daudz strādāt ar domēnu nosaukumu konfigurāciju, kuras ērtums ir atkarīgs no pakalpojuma sniedzēja izveidotās sistēmas. Šajā nodaļā tiek apskatīts, kāds ir 3 aplūkojamo mākoņskaitļošanas pakalpojumu sniedzēju piedāvātais DNS serviss un kādas ir to priekšrocības pār citiem.

*AWS* piedāvātais serviss saucas *Route 53* [ARO 17]. *Route 53* savieno lietotāju pieprasījumus ar *AWS* infrastruktūru, kā, piemēram, *EC2* instancēm, *Elastic Load Balancer* slodzes līdzsvarotājiem vai *Amazon S3* datu noliktavām. *Route 53* var arī tikt izmantots, lai novirzītu lietotājus uz infrastruktūru ārpus *AWS*. Tas var tikt izmantots, lai konfigurētu DNS statusa pārbaudes, lai novirzītu trafiku uz veselajiem galapunktiem vai patstāvīgi novērotu lietotņu statusu. *Route 53* piedāvā arī domēnu nosaukumu reģistrāciju un iespēju automātiski konfigurēt DNS iestatījumus, kā arī DNS slodzes līdzsvarošanu ar iespēju pielāgot svarus klienta DNS ierakstiem, kas nosaka, cik liela daļa no trafika tiek novirzīta kādiem galapunktiem.

*Microsoft* piedāvātais serviss saucas *Azure DNS* [ADO 17]. Tas ļauj pārvaldīt klienta DNS ierakstus ar tādu pašu pierakstīšanās informāciju, apmaksas sistēmu un atbalsta programmas līgumu, kā citiem *Azure* pakalpojumiem. *Azure DNS* vienīgajam no aplūkotajiem nav pieejama programmsaskarne, kas atsevišķās sistēmās var būt nozīmīgi, ja nepieciešams veidot pielāgotu DNS ierakstu interfeisu.

*Google Cloud Platform* piedāvātais rīks saucas *Google Cloud DNS* [CDO 17]. Tas ir mērogojams serviss, kurš darbojas uz tādu pašu infrastruktūru, uz kādu darbojas *Google* servisi. *Cloud DNS* spēj nodrošināt darbību, veidojot vai rediģējot miljoniem DNS ierakstu. DNS ierakstus iespējams pārvaldīt izmantojot gan tīmekļa saskarni, gan komandrindu. *Google* piedāvā arī REST programmsaskarni, kas klientiem ļauj izveidot sev pielāgotu DNS interfeisu.

No aplūkotajiem pakalpojumiem tīmekļa vietnē *Stackshare* visaugstāko vērtējumu ieguvis *Route 53*. Otrs augstākais vērtējums ir *Google Cloud Platform* pakalpojumam, taču

*Azure DNS* servisu neviens vietnes lietotājs nav pievienojis. Pēc tā var secināt, ka komūna par visnoderīgāko no aplūkotajiem uzskata *Route 53*.

### 3.7. Automātiskas mērogojamības iespējas

Automātiska mērogošana ļauj uzturēt sistēmas pieejamību, palielinot vai samazinot izmantotos mākoņskaitļošanas resursus, atkarībā no sistēmas noslogojuma vai citiem klienta uzstādītiem nosacījumiem. Multireģionālu sistēmu uzturēšanā nereti nākas strādāt ar lielu skaitu instanču, kas ievērojami sadārdzina pakalpojuma izmaksas, taču, izmantojot automātisko mērogošanu, iespējams tās samazināt.

*AWS* automātiskās mērogojamības [ASP 17] iestatījumos iespējams norādīt tāds nosacījumus, kā piemēram, pie kādas procesora vidējās noslodzes samazināt vai palielināt instanču skaitu. Iespējams uzstādīt minimālo, maksimālo un vēlamu instanču skaitu, kā arī zonu, kurā veidot jaunās instances, ja konkrētai sistēmai ir svarīga ģeogrāfiskā lokācija, kur instances darbojas. Ja sistēmai ir paredzamas noslodzes izmaiņas, piemēram, darba dienu rītos vai brīvdienās, ir iespējams izveidot grafiku, kas plānotu laicīgi aktivitātes pieaugumiem vai samazinājumiem. Iespējams izmantot arī *Amazon CloudWatch* monitoringa rīku, lai sūtītu trauksmes signālus, kas aktivizētu mērogošanas darbību.

*Azure* automātiskā mērogošana [AUT 17] saņem metrikas no resursiem un padod tās nosacījumiem, kurus klients uzstādījis. *Azure* sniedz iespēju izveidot sarežģītus, pārklājošus nosacījumus, lai funkcionalitāte būtu maksimāli pielāgojama. Tie var būt uz metrikām bāzēti, piemēram, pie noteikta procesora vai cietā diska noslodzes, kā arī uz laiku balstīti, piemēram, konkrētas dienas rīts. *Azure*, līdzās virtuālo mašīnu mērogošanai uz leju vai uz augšu, sniedz iespēju arī informēt noteiktus cilvēkus ar e-pastu palīdzību, piemēram, par noslodzes statusu un rīcības plānu, kas tiks izpildīts. Papildus tam, iespējams izmantot arī aizķeres, lai aktivizētu vairākas kompleksas darbības iekš vai ārpus *Azure* vai, piemēram, veiktu ziņojumus uz ārējiem resursiem kā *Slack* vai *Trello*.

Lai *Google Cloud* platformā uzstādītu automātisku mērogošanu, jāizvēlas pēc kāda kritērija un pie kādas vērtības mērogošana tiks veikta. *Google Cloud* piedāvā 4 veidu [AGO 17] kritērijus:

- vidējais procesora noslogojums;
- *Stackdriver* monitoringa metrikas - automātisko mērogošanu iespējams uzstādīt tā, ka tiek ievākti dati par specifisku metriku un veic mērogošanu, balstoties uz vēlamu noslogojumu. Iespējams izmantot arī klienta veidotas metrikas;

- HTTP slodzes līdzsvarotāja aktivitāte, ko var balstīt uz patēriņu vai pieprasījumiem sekundē;
- *Google Cloud Pub/Sub* rindas slodzi.

Aplūkojot visus 3 piedāvātos pakalpojumus, var secināt, ka automātiskās mērogošanas jomā visplašākās iespējas piedāvā *AWS* un *Azure*, kamēr *Google Cloud* risinājums ir relatīvi vienkāršāks, taču nosedz lielāko daļu gadījumu, kuri klientam var būt nepieciešami. Tas var tikt attiecināts uz to, ka *Google Cloud* no apskatītajiem 3 tirgū ir jaunākais. Svarīgs aspekts, ko ņemt vērā ir arī pieejamie mērogošanas veidi: gan *Azure*, gan *AWS* piedāvā vertikālu (esošo instanču resursu palielināšana vai samazināšana) un horizontālu (līdzīgu instanču pievienošana vai noņemšana) automātisko mērogošanu, kamēr *Google Cloud* tikai horizontālu.

### 3.8. Automātiska izvietošana

Automātiska izvietošana programmatūras izstrādes industrijā kļūst par standarta praksi un jo īpaši svarīgi tas kļūst, kad sistēma tiek darbināta uz lielu daudzumu instanču. Automātiskai izvietošanai ir vairums priekšrocību:

- zemāka kļūdu iespējamība;
- lai veiktu izvietošanu, nav nepieciešamas nekādas instrukcijas vai pamācības;
- laika ietaupījums;
- iespēja biežāk veikt sistēmas atjauninājumus.

*AWS* piedāvātais izvietošanas rīks saucas *CodeDeploy* [ACD 17]. *CodeDeploy* pilnībā automatizē lietotnes izvietošanu izstrādes, testa un produkcijas vidēm. Izvietošanas laikā *CodeDeploy* veic secīgu atjaunināšanu uz *EC2* instancēm. Ir iespējams norādīt, cik instanču atjaunināšanas laikā drīkst vienlaicīgi tikt noņemtas no tiešsaistes. *AWS* piedāvā divu veidu izvietošanas:

- *blue/green* - tiek izveidotas papildus instances, uz tām tiek veikti atjauninājumi; pēc tam, kad veiktas nepieciešamās pārbaudes un testi, trafiks tiek manuāli vai automātiski novirzīts uz jauno versiju instancēm. Pēc noteikta laika iepriekšējās instances tiek noņemtas;
- *in-place* - lietotne uz katras izvietošanas grupas instances tiek apturēta, tiek uzstādīta jaunākā lietotnes versija, lietotne tiek iespējota un apstiprināta.

Izmantojot *CodeDeploy*, iespējams automātiski vai manuāli apturēt un atgriezt izmaiņas, ja izvietošanas procesā radušās kļūdas, sekot līdz izvietošanas procesam iespējams

caur *CodeDeploy* tīmekļa saskarni vai komandrindu. Rīks nav atkarīgs no izmantotās platformas un darbojas ar jebkāda veida lietotnēm.

*Microsoft Azure* nav specifiska rīka, kas būtu paredzēts automātiskai izvietošanai, taču tā ir daļa no *Azure App Service* funkcionalitātes kopas [CDA 16]. *App Service* atbalsta trīs dažādus izvietošanas procesus:

- tieša failu augšuplāde caur FTP vai FTPS, izmantojot klientu, kā, piemēram, *FileZilla*. *App Service* šādā gadījumā nepiedāvā nekādu papildus funkcionalitāti, kā versiju kontroli vai failu pārvaldību;
- izmantojot *Kudu* dzini. Tas atbalsta trīs dažādus izvietošanas veidus:
  - satura sinhronizēšana no *OneDrive* vai *Dropbox*;
  - uz repozitoriju bāzēta nepārtraukta izvietošana, sinhronizējot ar servisiem *GitHub*, *Bitbucket* vai *Visual Studio Team Services*;
  - uz repozitoriju bāzēta izvietošana, manuāli sinhronizējot ar lokālu *Git* repozitoriju;
- tīmekļa izvietošana. Programmkods tiek izvietots *App Service*, izmantojot kādu no *Microsoft* rīkiem, piemēram, *Visual Studio*. Salīdzinājumā ar *Kudu* pieeju, ar šo metodi lietotņu binārie faili tiek uzbūvēti pirms izvietošanas *Azure*. Līdzīgi kā ar FTP pieeju, arī izmantojot šo pieeju, nekādu citu papildus funkcionalitāti *App Service* nenodrošina.

*Google Cloud* piedāvātais serviss saucas *Deployment Manager* [CDM 16]. To iespējams izmantot, lai veidotu elastīgas veidnes, kurās ir nodefinētas izvietošanas konfigurācijas un ar kurām var veikt izvietošanu uz tādiem servisiem kā *Google Cloud Storage*, *Google Compute Engine* un *Google Cloud SQL*. *Deployment Manager* piedāvā tādas iespējas [BLA 16], kā

- izvietot uz vairākiem resursiem paralēli;
- pievienot, dzēst vai mainīt resursus izvietošanas laikā;
- redzēt priekšskatījumu izmaiņām, ko veiks *Deployment Manager* pirms tās apstiprināt;
- izvietošanas veidnēs padot parametrus kā zonu, instanču skaitu, vidi un saņemt atgrieztās vērtības kā piešķirtās IP adreses, saites uz instancēm u.c.
- resursiem pievienot references uz citiem resursiem, tādejādi veidojot atkarības un kontrolējot resursu izveides kārtību.

Katram no apskatītajiem rīkiem ir savas priekšrocības, taču arī šajā jomā *AWS* piedāvātais rīks piedāvā visplašākās iespējas, būtiski nesarežģot tā izmantošanu. Arī komūnas vērtējums vietnē *StackShare* liecina par to, ka lietotājiem šis rīks ir parocīgs.

### 3.9. Pakalpojuma izmaksas

Multireģionālu sistēmu uzturēšanā svarīgs aspekts mākoņskaitļošanas pakalpojumu izvēlē ir cena. Industrijā ir liela konkurence un cenas pēdējos gados ir krasi kritušās, piemēram, *AWS* pēdējo sešu gadu laikā ir samazinājuši cenas 44 reizes [KIM 14], arī *Microsoft* un *Google* vairākas reizes samazinājuši cenas, mēģinot konkurēt ar *AWS*. Tomēr, sistēmai darbojoties uz daudzām virtuālajām mašīnām, uzturēšanas izmaksas var būtiski ietekmēt klienta finanses.

Šajā salīdzinājumā netiek ņemtas vērā dažādas akcijas un atlaides, ko piedāvā pakalpojumu sniedzēji, jo tās mēdz būt neilglaicīgas un var ātri kļūt neaktuālas, tomēr tās būtu jāņem vērā veicot savu izvēli, jo atsevišķos gadījumos iespējams ietaupīt pat vairāk kā 25% no līdzekļiem

3.2. tabula

Mākoņskaitļošanas pakalpojumu sniedzēju izcenojumu salīdzinājums

Resursa tips	<i>AWS</i> instance	<i>Azure</i> instance	<i>Google</i> instance	<i>AWS</i> stunda s likme	<i>Azure</i> stunda s likme	<i>Google</i> stunda s likme	<i>AWS</i> cena/ GB RAM	<i>Azure</i> cena/ GB RAM	<i>Google</i> cena/ GB RAM
2 CPU ar SSD - standarta	m3.large	D2 v2	n1-standard-2	\$0.133	\$0.144	\$0.212	\$0.017	\$0.016	\$0.028
2 CPU ar SSD - ar papildus atmiņu	r3.large	D11 v2	n1-highmem-2	\$0.166	\$0.149	\$0.238	\$0.011	\$0.011	\$0.018
2 CPU ar SSD - ar papildus skaitļošanas jaudu	c3.large	F2	n1-highcpu-2	\$0.105	\$0.099	\$0.188	\$0.028	\$0.025	\$0.104
2 CPU bez SSD - standarta	m4.large	D2 v2	n1-standard-2	\$0.108	\$0.144	\$0.100	\$0.014	\$0.016	\$0.013
2 CPU bez SSD - ar papildus atmiņu	r4.large	D11 v2	n1-highmem-2	\$0.133	\$0.149	\$0.126	\$0.009	\$0.011	\$0.010
2 CPU bez SSD - ar papildus skaitļošanas jaudu	c4.large	F2	n1-highcpu-2	\$0.105	\$0.099	\$0.076	\$0.027	\$0.025	\$0.042

Pēc tabulas 3.2. [WEI 16] var secināt, ka *Google Cloud* piedāvā zemāko cenu 3 gadījumos un augstāko cenu 7 gadījumos. *Google Cloud* lielākoties ir zemākas cenas, kad netiek izmantots SSD un augstākās, kad tiek izmantots, jo lokāla 375 GB SSD pievienošana *Google* prasa daudz līdzekļu. Tāpat arī *Google* cenas ir augstākas, kad tiek aplūkota cena par GB operatīvās atmiņas uz paaugstinātas skaitļošanas jaudas instancēm, jo minētajām

instancēm ir mazāk nekā puse no operatīvās atmiņas, ko piedāvā *AWS* un *Azure* analogās instancēs.

*Azure* piedāvā zemāko cenu sešos no apskatītajiem gadījumiem un augstāko cenu četros no tiem un, salīdzinājumā ar *AWS*, *Azure* cenas ir tādas pašas vai zemākas 7 no 12 gadījumiem. *AWS* piedāvā zemāko cenu trīs gadījumos un augstāko cenu vienā no apskatītajām situācijām, kas nozīmē, ka lielākoties *AWS* cenas ir starp *Azure* un *Google* piedāvātajām.

### 3.10. Tehniskais atbalsts

Izmantojot uz mākoņa bāzētus pakalpojumus, klients savus datus uztic pakalpojuma sniedzēja serveriem. Atbilstoši pakalpojumu sniedzējam būtu jānodrošina pārstāvis, ar kuru klients var sazināties, ja tāda vajadzība rodas. Lai arī tiešsaistes komūnas aug un lielu daļu risinājumu iespējams atrast pašrocīgi, var gadīties sastapties ar problēmām, kuru dēļ nākas sazināties ar pakalpojumu sniedzēju.

Visi trīs apskatāmie pakalpojumu sniedzēji piedāvā tehniskā atbalsta iespējas, un objektīvākam salīdzinājumam tiks apskatīti divi atbalsta līmeņi: bezmaksas atbalsts, lai noskaidrotu, kādas iespējas ir bez papildus samaksas, kā arī lētākais pieejamais atbalsta līmenis, jo šajā līmenī izcenojums pakalpojumu sniedzējiem ir vislīdzīgākais.

3.3. tabula

Mākoņskaitļošanas pakalpojumu sniedzēju bezmaksas tehniskā atbalsta plāni

<i>AWS</i>	<i>Azure</i>	<i>Google Cloud</i>
<ul style="list-style-type: none"> <li>• Diennakts pieeja klientu kontu atbalstam, dokumentācijai un atbalsta forumiem.</li> <li>• Pieeja 4 galvenajām <i>AWS Trusted Advisor</i> programmas daļām.</li> <li>• Pieeja infrastruktūras statusa mērinstrumentu panelim.</li> </ul>	<ul style="list-style-type: none"> <li>• Pieeja <i>Azure</i> forumiem.</li> <li>• Iespēja rakstīt <i>Azure</i> mikroblogošanas vietnē <i>Twitter</i>.</li> <li>• Tehniskais atbalsts <i>Azure</i> pakalpojumu problēmām, izmantojot rīku <i>Resource Health</i>.</li> <li>• Diennakts atbalsts saistībā ar norēķiniem un lietotāju kontu problēmām.</li> </ul>	<ul style="list-style-type: none"> <li>• Tehniskais atbalsts bezmaksas izmēģinājuma laikā.</li> <li>• Pieeja <i>Google Help Center</i>.</li> <li>• Pieeja dokumentācijai un pamācībām.</li> <li>• Iespēja meklēt atbalstu <i>Google Groups</i> komūnā.</li> <li>• Diennakts atbalsts saistībā ar norēķiniem un lietotāju kontiem.</li> </ul>

Pēc tabulā 3.3. [ASU 17, AZS 17, SUP 17] apkopotās informācijas, var secināt, ka ir iespēja saņemt atbalstu, tai skaitā arī tehnisko atbalstu, neieguldot papildus līdzekļus. Visiem apskatītajiem pakalpojumu sniedzējiem ir pieejami rīki tehniskā atbalsta saņemšanai, kas neprasa cilvēkresursu no pakalpojuma sniedzēju puses. Papildus tam *Google Cloud* piedāvā tehnisko atbalstu, komunicējot ar *Google* pārstāvi, bezmaksas izmēģinājuma laikā, kas ir 12 mēneši.

#### 3.4. tabula

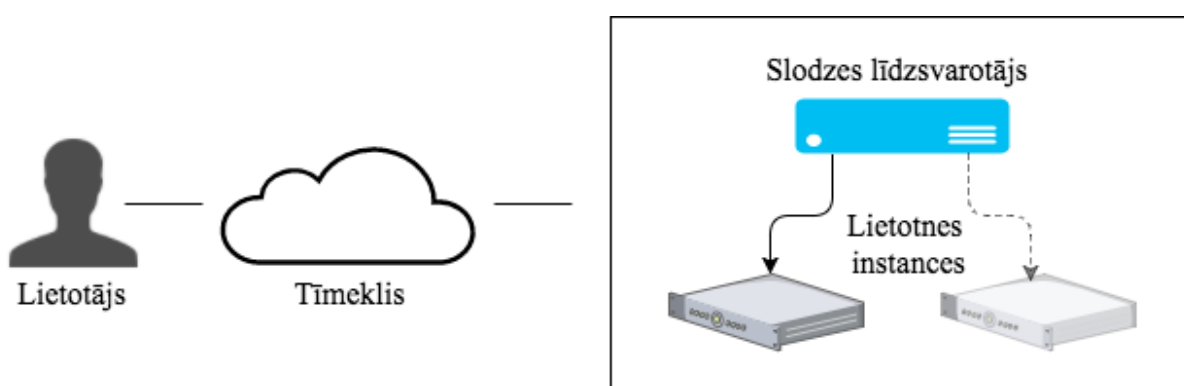
#### Mākoņskaitļošanas pakalpojumu sniedzēju pamata maksas tehniskā atbalsta plāni

<i>AWS Developer</i> plāns	<i>Azure Developer</i> plāns	<i>Google Cloud Silver</i> plāns
<p>Papildus bezmaksas plānam ir iespēja sazināties ar <i>Azure</i> pārstāvjiem ar sekojošiem nosacījumiem:</p> <ul style="list-style-type: none"> <li>• cena: \$29/mēnesī,</li> <li>• sākotnējās atbildes laiks: līdz 12 stundām,</li> <li>• atbalsts tikai darba laika stundās,</li> <li>• viens pārstāvis, kurš var veidot neierobežota skaita atbalsta prasību,</li> <li>• iespējams uzstādīt tikai zemāko problēmas nopietnības līmeni,</li> <li>• arhitektūras atbalsts tikai pamata līmenī.</li> </ul>	<p>Papildus bezmaksas plānam ir iespēja sazināties ar <i>Azure</i> pārstāvjiem ar sekojošiem nosacījumiem:</p> <ul style="list-style-type: none"> <li>• cena: \$29/mēnesī,</li> <li>• sākotnējās atbildes laiks: zem 8 stundām,</li> <li>• atbalsts tikai darba laikā,</li> <li>• iespējams uzstādīt tikai zemāko problēmas nopietnības līmeni.</li> </ul>	<p>Papildus bezmaksas plānam ir iespēja sazināties ar <i>Google Cloud</i> pārstāvjiem ar sekojošiem nosacījumiem:</p> <ul style="list-style-type: none"> <li>• cena: \$150/mēnesī,</li> <li>• sākotnējās atbildes laiks: zem 4 stundām,</li> <li>• divi uzņēmuma pārstāvji, kuri var veidot atbalsta prasības,</li> <li>• atbalsts tikai darba laikā,</li> <li>• atbalsts saistībā ar arhitektūru tikai pamata līmenī.</li> </ul>

Pēc tabulā 3.4. [ASU 17, AZS 17, SUP 17] apkopotās informācijas var secināt, ka visi piedāvājumi ir līdzīgi, taču *Google Cloud* piedāvājums ir vairāk kā 5 reizes dārgāks par pārējo divu pakalpojumu sniedzēju plāniem. Šajā cenu līmenī *Google* spēj nodrošināt klientus ar divas līdz trīs reizes ātrāku sākotnējās atbildes laiku, kā arī sniedz iespēju diviem uzņēmuma pārstāvjiem veidot atbalsta prasības, kamēr *AWS* un *Azure* pieejams tikai viens pārstāvis. Visi no apskatītajiem pakalpojumu sniedzējiem šajā kategorijā piedāvā tehnisko atbalstu tikai darba dienās darba laikā.

### 3.11. Slodzes līdzsvarošanas iespējas

Slodzes līdzsvarošana [AND 17] ir tehnika, kas ļauj efektīvi novirzīt ienākošo tīmekļa trafiku uz vairākiem serveriem. Modernas tīmekļa lietotnes apkalpo simtiem un atsevišķos gadījumos pat miljoniem vienlaicīgu klientu pieprasījumu un prasītos datus atgriež relatīvi ātri. Augstas noslodzes lietotņu mērogošanai visbiežāk ir nepieciešams pievienot papildus serverus. Slodzes līdzsvarotājs kalpo kā starpnieks starp klientu un serveriem un ienākošo trafiku līdzīgā apjomā novirza uz visiem serveriem, tādējādi tiek maksimizēta esošo resursu efektivitāte un tiek nodrošināts, ka neviens serveris netiek pārlietu noslogots.



3.1. att. Slodzes līdzsvarotāja darbības diagramma

Slodzes līdzsvarotāja darbības princips redzams attēlā 3.1.

Multireģionālu sistēmu risinājumos katram reģionam ir nepieciešams savs slodzes līdzsvarotājs, tāpēc svarīgi aplūkot pakalpojumu sniedzēju piedāvātās slodzes līdzsvarošanas iespējas. Pieejami arī vairāki trešo pušu risinājumi, kuri šī darba ietvaros netiek apskatīti.

*AWS* piedāvātais slodzes līdzsvarošanas rīks ir *Elastic Load Balancing*. *Elastic Load Balancing* piedāvā divu veidu slodzes līdzsvarotājus, no kuriem abi nodrošina augstu pieejamību un automātisku mērogojamību. Pirmais no tiem ir klasisks slodzes līdzsvarotājs [CLB 17], kurš novirza trafiku balstoties uz lietotnes vai tīkla līmeņa informācijas, otrs – *Application Load Balancer*, kurš trafiku novirza balstoties uz specifisku lietotnes līmeņa informāciju, tai skaitā arī pieprasījuma saturu. Pirmais no nosauktajiem ir paredzēts primitīvai slodzes līdzsvarošanai starp EC2 virtuālajām mašīnām, kamēr *Application Load Balancer* var tikt izmantots, lai novirzītu trafiku uz dažādiem servisiem vai izlīdzinātu noslodzi starp dažādiem portiem uz vienas EC2 virtuālās mašīnas. *AWS Classic Load Balancer* atbalsta arī iespēju, izmantojot sīkfailus, piesaistīt lietotāju sesijas noteiktām EC2 virtuālajām mašīnām.

Lietotājam turpinot lietotni izmantot, jauni pieprasījumi tiek novirzīti uz to pašu instanci, kur iepriekš.

*Microsoft Azure Load Balancer* piedāvā tīkla līmeņa slodzes līdzsvarošanu, izdalot trafiku darbaspējīgās virtuālajās mašīnās. *Azure Load Balancer* var tikt nokonfigurēts sekojošos veidos:

- līdzsvarojošā ienākošā tīmekļa trafika slodzi uz virtuālajām mašīnām;
- līdzsvarojošā slodzi starp virtuālajām mašīnām virtuālā tīklā, starp virtuālajām mašīnām mākoņa pakalpojumos vai starp lokālām darbstacijām un virtuālām mašīnām (dalītā slodzes līdzsvarošana);
- novirzot ārējo trafiku uz noteiktu virtuālo mašīnu.

*Azure* slodzes līdzsvarotājs spēj dinamiski mainīt savus iestatījumus, kad instances tiek mērogotas uz augšu vai uz leju.

*Google Cloud Platform* piedāvā 3 veidu slodzes līdzsvarošanu:

- globālu ārēju slodzes līdzsvarošanu – līdzsvaro HTTP(S) trafiku, SSL trafiku vai TCP trafiku balstoties uz lietotāja tuvumu, pieprasījuma URL vai abiem;
- reģionālu ārēju slodzes līdzsvarošanu – tīkla trafiks tiek sadalīts instanču kopai reģionā; iespējams līdzsvarot jebkāda veida TCP/UDP trafiku;
- reģionālu iekšēju slodzes līdzsvarošanu – sadala trafiku no *Google Cloud Platform* virtuālajām mašīnām uz instanču grupu tajā pašā reģionā.

*Google Cloud Platform* piedāvā arī iespēju reģistrēt datus par visiem slodzes līdzsvarošanas pieprasījumiem, kas sūtīti uz slodzes līdzsvarotāju. Žurnāli var tikt izmantoti atklūdošanai, kā arī lietotāju trafika analizēšanai. Tos iespējams eksportēt un analizēt citos *Google* servisos: *Google BigQuery* un *Google Cloud Pub/Sub*.

Pēc veiktā pētījuma var secināt, ka pakalpojumu sniedzēju sniegtās iespējas galvenokārt ir līdzīgas. Atšķirības rodas, aplūkojot sekundārās funkcionalitātes, kur, piemēram, *AWS* sniedz iespēju izmantot sīkfailus, lai piesaistītu lietotāju noteiktai instancei. Apsverot slodzes līdzsvarošanas faktoru, tieši šīs detaļas ir vērts ņemt vērā, jo lielākajai daļai sistēmu pamata funkcionalitāti spēs nodrošināt visi pētījumā aplūkoto pakalpojumu sniedzēji. Vērts atzīmēt arī, ka *AWS* slodzes līdzsvarotājs tīmekļa vietnē *StackShare* novērtēts ar 34 balsīm, kamēr *Azure* un *Cloud Platform* nav vērtējuma, jo servisi nav vietnē pievienoti.

### 3.12. Mākoņglabāšana

Mākoņglabāšana [CST 17] ir datu uzglabāšanas veids, kur digitāli dati tiek glabāti loģiskās kopās. Fiziskā krātuve sastāv no vairākiem serveriem, atsevišķos gadījumos vairākās

lokācijās, un to parasti pārvalda satura mitināšanas uzņēmums. Mākoņkrātuvju nodrošinātāji ir atbildīgi par datu pieejamību un fiziskās serveru vides drošību. Organizācijām un cilvēkiem ir iespēja iegādāties vai nomāt atmiņu datu krātuvēs. Mākoņkrātuves ir nozīmīga mākoņskaitļošanas daļa. Glabāšanas risinājumus izmanto tādas tehnoloģijas kā lieli dati, lietu internets, datu bāzes, datu noliktavas u.c. Mākoņglabāšana tipiski ir uzticamāka, vieglāk mērogojama un drošāka nekā tradicionāli lokāli datu glabāšanas risinājumi. Arī multireģionālu sistēmu izstrādē un uzturēšanā mākoņglabāšana sniedz vairākas priekšrocības:

- nav nepieciešams iegādāties un uzturēt savu datu glabāšanas infrastruktūru dažādās pasaules vietās;
- daļu atbildības par datu drošību nes mākoņglabāšanas pakalpojumu sniedzējs;
- piedāvātie datu uzglabāšanas risinājumi ir pārdomāti un industrijas ekspertu izstrādāti, kas izstrādātājam ļauj vairāk savu resursu veltīt sistēmas izstrādei, nevis datu glabāšanas nodrošinājumam.

Šī pētījuma ietvaros tiek aplūkoti mākoņskaitļošanas pakalpojumu sniedzēju bloku tipa glabāšanas un objektu tipa glabāšanas piedāvātie mākoņglabāšanas risinājumi.

*AWS* atmiņas bloku glabāšanas risinājums ir *Amazon Elastic Block Store* [AEB 17]. Pakalpojums sniedz iespēju glabāt datus gan SSD, gan cietajos diskos. Tas nodrošina 99.999% pieejamību un automātiski kopē sējumu attiecīgajā reģionā, kas pasargā no datu zuduma cietā diska atteices gadījumā. EBS piedāvā iespēju veidot momentuzņēmumus un noglabāt *Amazon S3* ilglaicīgai glabāšanai. Tāpat ir iespēja dinamiski palielināt atmiņas lielumu, regulēt veiktspēju un mainīt EBS sējumu tipu, neapturot to darbību.

*AWS* piedāvātais objektu glabāšanas risinājums ir *Simple Storage Solution (S3)* [AMS 17]. Tas nodrošina 99,99999999% uzticamību un 99,99% pieejamību. Nepieciešamās atmiņas lielumu iespējams mērogot gandrīz neierobežotā daudzumā. *S3* piedāvā trīs veidu migrācijas iespējas – tīmekļa optimizētu, uz fiziskiem diskem bāzētu un trešo pušu savienojumu metodi.

*Google* bloku un objektu tipa glabāšanas risinājumi ir daļa no servisa ar nosaukumu *Google Cloud Storage* [CSG 17]. Tā piedāvātās iespējas ir ļoti līdzīgas *AWS. Cloud Storage* multireģionālai glabāšanai nodrošina pieejamību virs 99,95%. Gan *AWS*, gan *Google Cloud* izmaksas veidojas no tīkla noslogojuma un datu daudzuma, kas katru mēnesi tiek uzglabāts. Papildus izmaksas veido arī API pieprasījumu daudzums. Abi pakalpojumu sniedzēji nodrošina iespēju saņemt paziņojumus par objektu dzēšanu, izveidi vai atjaunošanu. *AWS* piedāvā 3 veidu ziņojumus uz citiem *AWS* servisiem: *Amazon Simple Notification Service*, *Amazon Simple Queue Service* un *AWS Lambda*. *Cloud Storage* paziņojumu funkcionalitāti risina ar aizķerēm.

*Microsoft Azure Storage* piedāvātais objektu glabāšanas risinājums saucas *Blob Storage* [IMA 17]. Izvēloties *Blob Storage*, objektus iespējams sadalīt aukstajā un siltajā glabātuvē, kas ļauj ietaupīt resursus, siltajā glabātuvē uzglabājot tikai tos resursus, kuriem nepieciešams bieži piekļūt. *Azure* izmaksas veidojas tikai no izmantotā atmiņas daudzuma, neatkarīgi no diska izmēra. *Azure Disk Storage* visiem diskem veido 2 rezerves kopijas, kas diska atteices gadījumā spēj pārņemt darbību. Pēc *Microsoft* pieejamās informācijas, *Disk Storage* izmanto unikālu kešdarbības arhitektūru, kas samazina datu nolases gaidīšanas laiku.

Salīdzinot tīmekļa vietnes *StackShare* [STA 17] komūnas vērtējumus aplūkotajiem mākoņglabāšanas pakalpojumiem, ievērojami labākais vērtējums ir *AWS* servisiem. S3 novērtēts ar 1630 balsīm un EBS novērtēts ar 78 balsīm. Tikmēr *Google* un *Microsoft* pakalpojumi novērtēti ar attiecīgi 46 un 30 balsīm.

## 4. GADĪJUMA IZPĒTE - ZIEDOJUMU PIESAISTES PLATFORMA *FUNDERFUL*

*Funderful* ir uzņēmums, kas strādā pie absolventu ziedojumu piesaistes rīka izveides ar tādu pašu nosaukumu *Funderful*, kurš tiek piedāvāts universitātēm visā pasaulē. Lai šo rīku veiksmīgi izmantotu, universitātes sistēmā importē datus par studentiem un absolventiem. Tā kā sniegtie dati ir sensitīvi un universitātes tiek pakļautas likumiem, kas katrā valstī var atšķirties, datus nepieciešams glabāt un apstrādāt uz serveriem, kas atrodas noteiktā reģionā. Nodaļā tiek apskatītas problēmas, ar kurām *Funderful* komanda sastopās, veidojot risinājumu, kas atbilst dažādu valstu likumdošanai, kā arī tiek aprakstīti problēmu risinājumi.

### 4.1. *Funderful* ziedojumu piesaistes platforma

Uzņēmums *Funderful* savu darbību sāka 2012. gadā, piedāvājot universitātēm, kas nodarbojas ar ziedojumu vākšanu, izveidot plaši pielāgojamas tīmekļa vietnes, kas paredzētas tieši šim mērķim. *Funderful* pakalpojumam ir raksturīgs spēliskošanas aspekts, vizuāli pievilcīgi infografiki un dizains, kā arī ērts ziedošanas process. Ar laiku uzņēmums ieguva starptautisku uzticību, un arvien vairāk augstākās izglītības iestādes izvēlējās sadarboties ar to.

Sākotnējā uzņēmuma darbībā katrai tīmekļa vietnei jeb ziedojumu vākšanas kampaņai tika veidota sava koda bāze. Izstrādē tika izmantotas *PHP* un *JavaScript* programmēšanas valodas un dati tika glabāti atvērtā pirmkoda relāciju datu bāžu sistēmā *MySQL*. Katrai jaunai kampaņai veidot savu koda bāzi kļuva neproduktīvi, jo viena projekta izstrādes laiks bija aptuveni 3 mēneši, kas neļāva uzņēmumam apkalpot vairāk klientu, bremsējot tā attīstību.

Lielā daļā kampaņu tika atrastas vairākas līdzības un tika meklēti kopsaucēji, kas kampaņas padarīja veiksmīgas. Šie kopsaucēji kļuva par pamatu jaunas sistēmas veidošanai, kas saīsinātu kampaņu izstrādes ilgumu. Jaunā sistēma tika rakstīta valodā *Ruby*, papildinot to ar *Ruby on Rails* ietvaru, un kā datu bāžu risinājums tika izmantots *MongoDB*. Iemesli šādām izvēlēm bija sekojoši:

- *Ruby on Rails* ir paredzēts ātrai izstrādei un ar savu objekt-orientēto vidi viegli uzņemt jaunas izmaiņas, tā ir atvērtā pirmkoda programmatūra ar lielu skaitu atvērtā pirmkoda bibliotēku;
- sistēmas plānošanas laikā *MongoDB* strauji auga popularitātē [DER 17] un pierādīja sevi kā spēcīgu konkurentu starp citām datu bāžu sistēmām, kā arī tam ir liels komūnas atbalsts.

Sākotnēji sistēma tika izvietota uz *IBM SoftLayer* serveriem, taču gadu vēlāk finansiālu atbildību dēļ 2016. gada sākumā tika migrēta uz *Microsoft Azure* serveriem. Gan *IBM*, gan *Microsoft* jaunuzņēmumiem piedāvā piedalīties programmās, kur pakalpojumu sniedzējs nodrošina uzņēmumu ar gandrīz neierobežotiem mākoņskaitļošanas resursiem uz pirmo apkalpošanas gadu (*Microsoft* gadījumā uz pirmajiem 3 apkalpošanas gadiem), lai palīdzētu uzņēmumam augt un potenciāli iegūstot sev jaunu klientu. Šis iespējamais uzņēmums *Funderful* izmantoja, lai ne tikai samazinātu savas sistēmas uzturēšanas izmaksas, bet arī atrastu uzņēmumam vispiemērotāko mākoņskaitļošanas pakalpojumu sniedzēju.

Sākotnējās sistēmas plānošanā netika ieguldīts pietiekami daudz laika un pārejas periods no patstāvīgiem projektiem uz sistēmu radīja virkni problēmu. Sistēma tika audzēta līdz ar klientu bieži mainīgajām un individuālajām prasībām, kas sistēmas uzturēšanu ilgtermiņā padarīja īpaši grūtu. Arī izvēle izmantot *MongoDB* izrādījās kļūdaina, jo relāciju datu bāzu sistēma būtu piemērotāka izstrādātajam risinājumam. Dziļāk izziņot klientu vajadzības, noskaidrojās arī, ka daļai klientu ir svarīgi, ka to dati tiek glabāti un apstrādāti viņiem atbilstošā reģionā. Piemēram, universitāte Vašingtonā uzstāj, lai dati par viņu absolventiem glabājas ASV robežās, bet universitātei Lielbritānijā ir svarīgi, ka dati glabājas Eiropas Savienībā. Esošā sistēma nebija veidota, lai apmierinātu šo prasību, un izmaiņas, ko būtu jāveic ar esošo arhitektūru būtu bijušas sarežģītas un laikietilpīgas. Pagaidu risinājums šai problēmai bija pēc vajadzības pārcelt instances no vienas *Microsoft Azure* zonas uz citu. Šīs un citu nepilnību dēļ 2016. gada beigās uzņēmuma IT komanda izvērtēja izveidotās sistēmas arhitektūras trūkumus un tika nolemts veidot jaunu sistēmu, kur šie trūkumi tiktu adresēti. Līdz ar jaunās sistēmas izstrādi tika nolemts mainīt mākoņskaitļošanas pakalpojumu sniedzēju. Lēmuma iemesli detalizētāk tiek aprakstīti turpmākajās apakšnodaļās.

## 4.2. Mākoņskaitļošanas pakalpojumu sniedzēja maiņas pamatojums

Uzņēmuma komandai *Microsoft Azure* gada laikā sagādāja vairākas problēmas, kas kļuva arī par iemeslu mainīt mākoņskaitļošanas pakalpojumu sniedzēju:

- kad *Azure* domēnu nosaukumu serviss pārstāja strādāt, *Microsoft* neziņoja par dīkstāvi 6 stundas un vienīgais veids, kā sazināties ar *Azure* pārstāvi bija caur mikroblogošanas vietni *Twitter*;
- *Funderful* sistēma kļuva nepieejama, jo *Azure* nebija norādīta neviena kredītkarte un vienīgais veids, kā to izdarīt ir caur interneta pārlūku *Internet Explorer*, kurš nevienam komandas biedram nebija pieejams, jo uzņēmumā

darbs noritēja uz datoriem ar *macOS* operētājsistēmu. Sistēma bija dīkstāvē 4 stundas;

- *Azure*, migrācijas laikā no iepriekšējā kontrolpaneļa uz jaunu, dažbrīd neviens no konkrēta servisa kontrolpaneļiem nebija pieejams, jo iepriekšējie kontrolpaneļi lietotāju pāradresēja uz jaunajiem kontrolpaneļiem, taču tie nestrādāja korekti;
- komandrindas saskarnes dokumentācija bija galvenokārt veidota priekš *CMD* (komandrindas saskarne *Windows* operētājsistēmā), kas apgrūtināja darbu strādājot ar *Unix* sistēmām;
- lielā daļā uzdevumu, kas saistīti ar infrastruktūras uzstādīšanu *Azure* vidē, prasītās vērtības nācās ievadīt manuāli, piemēram, datu bāzu komplektēšanas ievadē nebija pieejamas iepriekš noteiktas vērtības. Līdzīgi arī ar *Linux* virtuālajām instancēm: sagatavi nebija iespējams izvēlēties no iepriekš noteiktām iespējām, un lai *Azure* atpazītu izvēli, tās nosaukums bija jākopē no citas vietnes;
- *Azure* administrācijas panelis komandai nelikās pietiekami intuitīvs un ērts. Reizēm visi mākoņresursi tika nestrukturēti izvietoti vienā skatā, citos skatos tie tika izdalīti pārāk detalizēti bez iespējas atrast līdzīgos;
- lielāko daļu uzdevumu, kas saistīti ar infrastruktūras konfigurāciju, nācās veikt no komandrindas saskarnes, bez iespējas tās mainīt no *Azure* administrācijas paneļa.

Neskatoties uz to, ka *Azure* jaunuzņēmumiem piedāvā bezmaksas resursus 3 gadu garumā, *Funderful* komanda izvēlējās mainīt mākoņskaitļošanas pakalpojumu sniedzēju jau pēc pirmā apkalpošanas gada.

Jauna mākoņskaitļošanas pakalpojuma sniedzēja izvēle var tikt balstīta uz ļoti atšķirīgiem faktoriem, piemēram, ātrums, drošība vai iespējams komūnas atbalsts, kuri atkarīgi no konkrētās sistēmas vai uzņēmuma. Uzņēmuma *Funderful* izvēle bija *AWS*, kas tika balstīta uz sekojošiem aspektiem:

- *AWS* ir finansiāli pretimnākoši jaunuzņēmumiem un veido dažādas akcijas, piemēram, *AWS Activate* [AAC 17], caur kurām uzņēmumi var saņemt iespēju *AWS* pakalpojumus izmantot bez maksas vai ar būtiski samazinātu cenu;
- *AWS* ir pieejams ievērojami plašāks komūnas atbalsts, kas apstiprinās arī šī pētījuma ietvaros. Komūnas atbalsts ir īpaši svarīgs mazām komandām bez lielas pieredzes, kā tas ir uzņēmumā *Funderful*;

- *AWS* ir plaša rīku un servisu bibliotēka [CPR 17], kas apmierina gandrīz visas klientu vajadzības, to skaitā 9 servisi saistīti ar mākoņskaitļošanu un 7 servisi, kas saistīti ar datu bāžu uzturēšanu;
- salīdzinot ar *Azure*, *AWS* piedāvā daudz ērtāku lietotāju un pieeju pārvaldību, kas ir nozīmīgi, kad daļai komandas vajag specifisku pieeju kādam servisam;

### 4.3. Jaunās *Funderful* sistēmas izstrāde un infrastruktūra

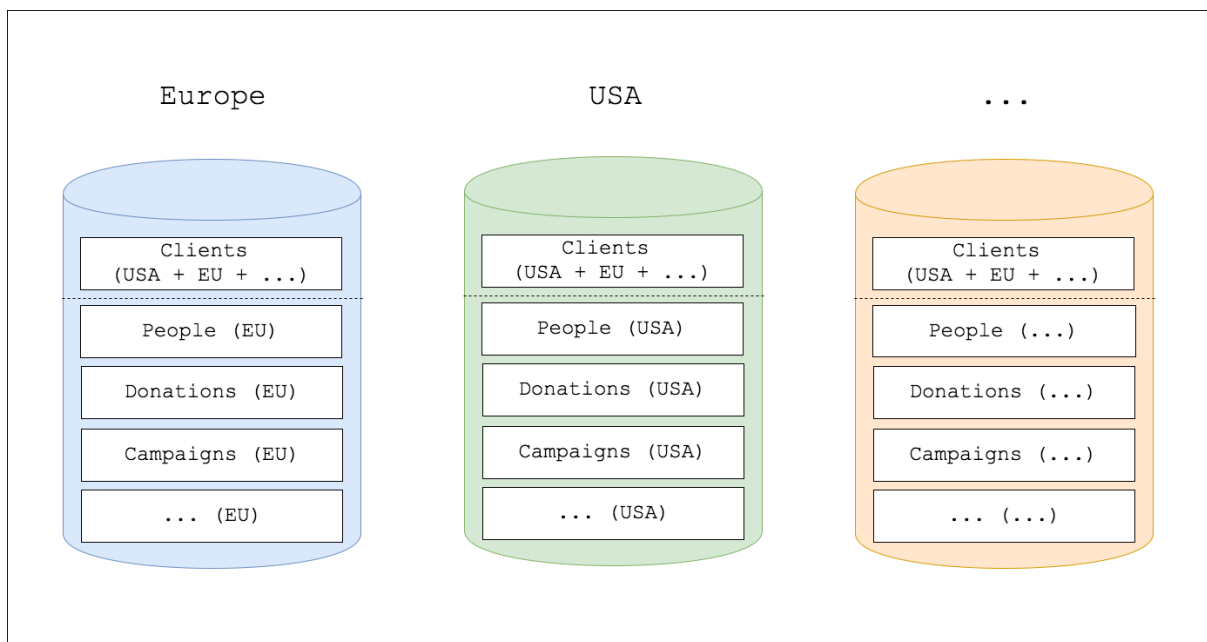
Jaunajā sistēmā tika nolemts turpināt izmantot *Ruby on Rails* satvaru, jo komandai jau bija salīdzinoši liela pieredze ar šo tehnoloģiju un nebija iemeslu to mainīt. Datu glabāšanai tika nolemts izmantot atvērtā pirmkoda relāciju datu bāžu sistēmu *MariaDB*, kuras popularitāte kopš 2013. gada ir mēreni augusi [DER 17]. Lai paātrinātu izstrādes gaitu, tika iegādāts gatavs klientu administrācijas paneļa risinājums, kurš veidots uz *AngularJS* satvara, to papildinot ar vairākām atvērtā pirmkoda bibliotēkām. Sākotnēji resursu trūkuma dēļ jaunajā sistēmā netika iestrādāta iespēja izvēlēties reģionu, kurā glabāt klienta datus. Funkcionalitāti tika ieviesta reizē ar migrāciju no *Azure* uz *AWS*.

*Funderful* sistēma glabā un apstrādā datus divos reģionos ar iespēju pievienot neierobežotu skaitu reģionu. ASV tirgum tiek izmantota “US East (N. Virginia)” un Eiropas tirgum “EU (Īrija)” zonas. Skaitļošanai tiek izmantots *AWS Elastic Compute Cloud (EC2)* un datu glabāšanai - *AWS Simple Storage Service (S3)*. Katram reģionam ir divas skaitļošanas virtuālās mašīnas. Sistēmas noslogojums ļautu iztikt ar vienu virtuālo mašīnu, taču šādā situācijā, vienas virtuālās mašīnas atteices gadījumā, otra turpinātu darbu un pilnībā nodrošinātu servisu. Šo *AWS* virtuālo mašīnu tips ir “t2.large”, tām ir 2 virtuālie procesori un 8GB atmiņa. Šādā tipa instances ir paredzētas darbam, kas bieži neizmanto pilnu procesora jaudu, taču brīžiem ir nepieciešama papildus jauda, piemēram, tīmekļa serveriem, testa vidēm, mikro servisiem vai noteikta tipa lietotnēm [AEC 17]. Tās ir ekonomiskas un ir ar balansētu skaitļošanas jaudu, atmiņu un tīmekļa resursiem. “T2” tipa instances atbalsta tikai *AWS EBS* krātuves, kuras arī risinājumā tiek izmantotas. Katram reģionam ir pieslēgts slodzes līdzsvarotājs, kurš novirza pieprasījumus balstoties uz virtuālo mašīnu procesoru noslogojumu.

### 4.4. Sistēmas arhitektūra

Attēlā 4.1. redzams vienkāršots sistēmas arhitektūras modelis. Katram reģionam ir sava datu bāze un katrā datu bāzē tiek uzturēti identiski tabulas “Clients” dati. Līdzīgās sistēmās

var arī iztikt bez vienojošās tabulas sinhronizācijas un izolēt pilnībā visus datus caur reģioniem, taču konkrētās sistēmas gadījumā tas bija nepieciešams.



#### 4.1. att. *Funderful* sistēmas arhitektūra

Sistēmas lietotājiem ar augstākajām piekļuves tiesībām, šajā gadījumā administratoriem, no kuriem visi ir *Funderful* komandas pārstāvji, jābūt iespējai autorizēties kā klientam vai zemāku piekļuvju lietotājam, lai, piemēram, palīdzētu klientam ar kampaņu konfigurāciju vai datu importu. Arī administratori, tāpat kā klienti, ir piesaistīti noteiktam reģionam, taču, lai varētu autorizēties, kā klients no cita reģiona, pašreizējā reģionā jābūt pieejai informācijai par klientiem, kas ir pārejos reģionos.

### 4.5. Reģionu datu bāžu datu sinhronizācija

Lai klientu dati visu reģionu datu bāzēs būtu sinhronizēti, nepieciešams veids kā piekļūt visu reģionu datu bāžu savienojumu iestatījumiem. Datu bāzes savienojuma iestatījumos tiek norādīti dati par to, kā saucas datu bāzes, ar kuru tiek veikts savienojums, kāds lietotājvārds un parole tiek izmantoti, datu bāzes saimniekdatora adrese un citi iestatījumi. Šādai funkcionalitātes izveidei iespējams izmantot inicializētājus. Tie ir programmkoda fragmenti, kas izpildās serverim sākot darbību. Šīs problēmas risinājumam inicializētājs tiek izmantots, lai atrastu visu reģionu datu bāžu savienojumu iestatījumus. Iegūtie reģionu iestatījumi tiek iterēti un katram reģionam tiek izveidota klase ar nosaukumu, kas atšķir reģionu no reģiona. Katrai reģiona klasei, izmantojot tai piesaistītos datu bāzes savienojuma iestatījumus, tiek

izveidots savienojums ar reģiona datu bāzi. Reģionu klases tiek izveidotas kā konstantes, kas padara tās pieejamas izsaukumiem no jebkuras citas klases. Šīs sistēmas ietvaros katram reģionam tiek veidota sava klase, jo *Ruby on Rails* nav izolētu pavedienu (no angļu val. "thread-safe"). Tas var radīt problēmas, kad 2 administratori mēģina piekļūt datiem. Piemēram, atlasot visu reģionu kampaņas, visi reģioni tiek iterēti un līdz ar tiem, mainās arī savienojums ar attiecīgo datu bāzi. Ja viena administratora datu atlase jau beigusī darbu ar pirmo reģionu un ir nomainījies savienojums ar datu bāzi, bet otra administratora datu atlasei vēl nepieciešams iepriekšējais savienojums, var gadīties, ka otram administratoram tiks atgriezta kļūda.

Tas nozīmē, ka, ja sistēmā autorizējušies 2 administratori un abi vēlas autorizēties kā klienti no atšķirīgiem reģioniem, tad pirmajam būs problēmas piekļūt attiecīgā reģiona datiem, kad autorizēsies otrs. Aprakstītā datu bāžu savienojumu iestatījumu iegūšanas risinājuma pirmkods pieejams 1. pielikumā.

Lai risinātu vienojošās tabulas, šajā gadījumā tabulas "clients", sinhronizāciju visos reģionos, brīdī, kad tabulā tiek veiktas kādas izmaiņas, tiek izsaukta funkcija, kas veic datu duplikāciju visos reģionos. Tajā tiek izveidots asociatīvs masīvs ar lauku nosaukumiem un vērtībām, kas tabulas ierakstam tiek mainītas vai pievienotas. Pēc tam tiek iterēti reģioni, kuri tiek izsaukti ar iepriekš aprakstītās konstantes palīdzību un katrā datu bāzē tiek veiktas asociatīvajā masīvā noteiktās izmaiņas. Līdzīgas darbības notiek arī tabulas ierakstu dzēšanas procesā, tiek iterēti reģioni un katrā datu bāzē konkrētais ieraksts tiek dzēsts. Risinājuma pirmkodu iespējams aplūkot 2. pielikumā.

#### **4.6. Aizķeru implementācija multireģionālā sistēmā**

*Funderful* sistēma plaši izmanto trešo pušu sistēmu integrācijas, lai nodrošinātu daļu sistēmas funkcionalitātes. Piemēram, tiešsaistes maksājumu apstrādes rīks *Stripe* [PST 17] vai e-pastu piegādes sistēma *SendGrid* [SES 17]. Šie trešo pušu rīki savu servisu nodrošināšanai izmanto aizķeres. Aizķeres ir HTTP(S) izsaukumi, kas tiek veikti, kad ir noticis kāds noteikts notikums, tiem parasti tiek izmantota POST metode. Piemēram, *Stripe* izmanto aizķeres, lai paziņotu citai sistēmai, ka mainījies maksājuma statuss vai klienta maksājumu kartei beidzies termiņš. *SendGrid*, savukārt, izmanto aizķeres, lai citai sistēmai, kas izmanto *SendGrid* integrāciju, paziņotu, ka e-pasts ir veiksmīgi piegādāts vai persona e-pastu ir atvērusi. Aizķeres sniedz iespējas, ko standarta programsaskarnes nespēj. Lai realizētu šo funkcionalitāti ar programsaskarni, trešās puses sistēmai būtu nepieciešams ik pēc noteikta laika intervāla sūtīt pieprasījumu uz pakalpojuma sniedzēja, piemēram, *SendGrid*, serveriem,

lai uzzinātu, vai persona ir atvērusi e-pastu vai nē. Šāda metode nespēj nodrošināt informācijas pieejamību reāllaikā un lieki noslogo pakalpojuma sniedzēju serverus.

Lai izmantotu trešo pušu sistēmu aizķeru sniegtās iespējas, pakalpojuma sniedzēja sistēmā jānorāda tīmekļa adrese, uz kuru dati tiks sūtīti, taču iespējams norādīt tikai vienu adresi, kas multireģionālā sistēmā rada sarežģījumus. Tā kā pakalpojums tiek izmantots visos reģionos, nepieciešams veids kā dati sasniegtu visus serverus. Viens no iespējamajiem variantiem, kā aizķeru integrāciju risināt multireģionālā sistēmā, ir katram 3. pušu pakalpojumu sniedzējam, kurš izmanto aizķeres, izveidot vienu gala punktu jeb URI, uz kuru pieprasījumus sūtīt. Kad tiek veikts pieprasījums uz attiecīgo URI, tiek izpildīta funkcija, kas pieprasījumu apstrādā.

Aizķeru pieprasījumi visbiežāk tiek asociēti ar kādu noteiktu resursu - izsūtītu e-pastu, izsūtītu e-pastu kopu, maksājumu, servisa abonementu u.c. Lai šo resursu varētu atrast, sistēmas datu bāzē jānoglabā identifikators, pēc kā atpazīt attiecīgo resursu. Kā piemērs tiek izmantots regulārs maksājums *Stripe* maksājumu sistēmā. Regulārs maksājums šajā kontekstā jāuztver kā abonementa veida maksājums, kas tiek veikts ik pēc noteikta laika intervāla. Lai sistēmā nonāktu dati par to, ka regulārais maksājums ir veiksmīgi veikts, *Stripe* sūta pieprasījumu sistēmai, un, kā vienu no parametriem, nodod šī abonementa un personas identifikatorus.

Izmantojot minētos vaicājuma parametrus, pirmā reģiona datu bāzē tiek meklēts attiecīgais abonementa datu bāzes ieraksts. Ja ieraksts tiek atrasts, vaicājums ir nonācis pie atbilstošā reģiona instances un tiek apstrādāts. Pretējā gadījumā tiek izpildīta funkcija, kas gadījumā, ja resurss nav piesaistīts reģionam, uz kuru saņemts pieprasījums, tas tiktu novirzīts uz nākošo reģionu. Tieša pāradresācija šajā gadījumā nav pieejama. Tā vietā no esošā pieprasījuma galvenes un ķermeņa tiek izveidots identisks pieprasījums, taču ar tīmekļa adresi, kas norāda uz citu reģionu. Darbība tad turpina izpildīties uz citas virtuālās mašīnas citā reģionā. Pirms tam gan jāveic pārbaude, vai visi nedefinētie reģioni jau nav mēģināti. Risinājuma pirmkods ir aplūkojams 3. pielikuma 3 attēlos.

Aizķeru problēmai iespējami arī citi risinājumi. Lielākā daļa pakalpojumu sniedzēju ļauj kopā ar resursu, datu bāzē noglabāt arī speciālas klientam raksturīgas vērtības. Iepriekš aplūkotajā regulāro maksājumu gadījumā būtu iespējams līdz ar abonementu *Stripe* sistēmā saglabāt vērtību, kas norādītu, kuram reģionam attiecīgais abonements ir piesaistīts. Kad ar šo abonementu tiek veiktas darbības un *Stripe* sūta aizķeri klienta sistēmai, sistēma atkarībā no reģiona vērtības novirza pieprasījumu uz attiecīgo reģionu. Šāds risinājums samazinātu maksimālo pāradresāciju skaitu līdz 1. Tāpat būtu iespējams arī izveidot tam atsevišķu virtuālo mašīnu, kas veiktu šīs pāradresācijas atkarībā no saņemtās vērtības. Šāda risinājums

trūkums ir tas, ka papildus virtuālā mašīna tērētu finansiālos resursus, kā arī būtu nepieciešams atsevišķs koda repozitorijs, kuru izmaiņu gadījumā vajadzētu atkārtoti izvietot uz virtuālās mašīnas. Pastāv arī risks, ka nākotnē var būt nepieciešams integrēt servisu, kas nesniedz iespēju savā sistēmā glabāt laukus un vērtības, ko klients vēlas. Tādā gadījumā vispraktiskākais risinājums ir pirmais no aprakstītajiem.

## 4.7. Multireģionālu sistēmu automātiska izvietošana

Automātiska izvietošana programmatūras izstrādes industrijā kļūst par standarta praksi. Tai ir virkne priekšrocību, kā zemāka kļūdu iespējamība, laika ietaupījums, iespēja biežāk veikt sistēmas atjauninājumus u.c. Multireģionālu sistēmu izvietošana uz mākoņskaitļošanas pakalpojumu sniedzēja virtuālajām mašīnām prasa papildus pielāgošanu.

*Funderful* automātiskai izvietošanai izmanto satvaru *Capistrano*. *Capistrano* ir satvars, kas paredzēts automatizētas izvietošanas skriptu veidošanai. Tas ir rakstīts valodā *Ruby*, taču to var izmantot, lai izvietotu projektus jebkurā valodā vai satvarā. *Capistrano* izvietošanas laikā izveido SSH savienojumu ar serveriem un izpilda soļus, kurus lietotājs ir nodefinējis. Izpildes soļus iespējams veidot pašrocīgi vai arī izmantot *Capistrano* kopienas veidotas uzdevumu bibliotēkas.

```
1. namespace :deploy do
2.   task :production do
3.     before 'deploy:starting', 'deploy:test_suite'
4.   end
5.
6.   after 'deploy:published', 'backup:perform_backup'
7.   after 'deploy:published', 'custom_migrations:execute'
8.
9.   # Configure database.yml for each region appropriately
10.  before 'deploy:setup_config', 'database:generate_database_yaml'
11.  before 'deploy:setup_config', 'database:symlink'
12.  after 'deploy:symlink:shared', 'database:symlink'
13.  [..]
14.
15.  # Clear public/deploy file cache storage on each of the regions
16.  after :deploy, 'clear_cache:remove_all'
17.  after :deploy, 'aws:create_image'
18.
19.  [..]
20.  after :finishing, 'deploy:cleanup'
21. end
```

### 4.2. att. Sistēmas automātiskās izvietošanas skripts

Attēlā 4.2. redzama daļa skripta, kas izpildās *Funderful* sistēmas izvietošanas laikā. Pirms tiek veiktas kādas citas darbības, tiek izpildīta automātisko testu kopa. Izvietošanas process tiek pārtraukts, ja kāds no testiem neizpildās korekti. Tas samazina iespēju, ka

produkcijas vidē sistēmas darbībā rodas kļūdas. Kā nākamā darbība tiek veikta datu bāzu rezerves kopiju izveide. Šajā gadījumā nepieciešams veidot pielāgotu dublēšanas funkciju, jo dublēšanu jāizpilda datu bāzēm visos reģionos, ko pēc noklusējuma neatbalsta izvēlētais izvietojuma satvars.

Lai šo funkcionalitāti realizētu, *Ruby* bibliotēka “EC2” izveido savienojumu ar *AWS* virtuālajām mašīnām un noskaidro, kurš serveris atbild par kādiem uzdevumiem. Lai tas notiktu, vispirms *AWS* virtuālajām mašīnām jāizveido atzīme, kam tā ir paredzēta. Tās tiek iterētas, tiek veikta papildus pārbaude, vai reģionā jau nav veikta dublēšana, kas var gadīties, ja *AWS* tiek kļūdas pēc izveidota instance ar neatbilstošu atzīmi, un tikai tad tiek veikta dublēšana. Aprakstīto darbību pirmkodu iespējams aplūkot 4. pielikumā.

Pēc visu reģionu datu bāzu dublēšanas tiek veikta datu bāzu migrācija. Arī šis process jāveic visās datu bāzēs, un tam nepieciešams pielāgots risinājums. Migrāciju izpildei tiek izsaukts līdzīgs skripts kā datu bāzu dublēšanas procesā, bet dublēšanas komanda tiek aizvietota ar migrāciju izpildes komandu.

Lai *Ruby on Rails* lietotne spētu komunicēt ar datu bāzi, katra reģiona instancēs, kas darbina šo lietotni, jāveic datu bāzes savienojuma iestatījumu faila konfigurēšana. Izstrādes vidē katram izstrādātājam šī datne var atšķirties, tāpēc tā neglabājas kopējā sistēmas repozitorijā. Savienojuma iestatījumu failā tiek norādīti tādi parametri kā datu bāzes adapteris, kas aplūkojamās sistēmas gadījumā ir “mysql2”, datu bāzes kodējums, nosaukums, lietotājvārds un parole, kas tiks izmantoti autorizācijai u.c. Uz katras *AWS* instances, kas darbina lietotni, datne atšķirsies, jo katrā no tiem jānorāda sava resursdatora adrese, datu bāzes nosaukums, kā arī autorizācijas lietotājvārds un parole. Skripts, kurš veic konfigurācijas failu izveidi un augšupielādi tiek izsaukts pēc datu bāzu migrāciju izpildes.

Sekojošās darbības tiek veiktas visām instancēm, kurām *AWS* ir norādīta attiecīgā loma, šajā gadījumā nepieciešamas instances, kuras darbina lietotnes serverus. No iepriekš sagatavotas veidnes, kura redzama 4. pielikumā attēlā 4.3., kur katram reģionam ir norādīta sava datu bāzes savienojuma konfigurācija, tiek izveidoti faili, kas tālāk tiek augšupielādēti katras instances attiecīgajā direktorijā. Pēc servera atkārtotas palaišanas, šī konfigurācija stājas spēkā un sistēma var turpināt darbu. Skripta izpildē tiek veiktas arī papildus darbības, kā *MySQL* lietotāju datnes ģenerēšana, kas ļauj izpildīt datu bāzu dublēšanas un izpildes beigās pagaidu datnes tiek dzēstas. Izvietojuma beigās tiek notīrīta kešatmiņa un pēc tam, kad ir sākusi darboties lietotnes jaunā versijā, vecās versijas mape tiek nodzēsta.

## 4.8. Gadījuma izpētes kopsavilkums

Papildus apskatītajām problēmsituācijām, izstrādātāju komanda sastapās arī ar citām sistēmai specifiskām problēmām, kuras šī pētījuma ietvaros netiek aplūkotas. Lai pēc iespējas samazinātu resursus, kas tiek patērēti sistēmas pielāgošanai multireģionālai datu glabāšanai un apstrādei, sistēmas plānošanas un izpētes periodam ieteicams veltīt vismaz 2 reizes vairāk laika, nekā tas būtu nepieciešams standarta sistēmai.

Šis laiks galvenokārt būtu jāvelta trešo pušu bibliotēku un pakalpojumu izpētei, jo nereti, tieši strādājot ar tiem, var nonākt situācijās, kur jāmeklē nestandarta risinājumi. Atvērtā pirmkoda trešo pušu bibliotēku un programmatūras gadījumā, ir iespējams tās rediģēt un pielāgot savām vajadzībām, taču ar servisiem, piemēram, mākoņskaitļošanas pakalpojumu sniedzējiem vai maksājumu operāciju nodrošinātājiem, gandrīz nekad tādas iespējas nebūs. Retos gadījumos pakalpojumu sniedzējs izpildītu šādas prasības, ja klienta prasītajai funkcionalitātei ir potenciāls būt noderīgai arī citiem klientiem.

## REZULTĀTI

Bakalaura darba “Mākoņskaitļošanas pakalpojumu izmantošana multireģionālu sistēmu uzturēšanai” mērķis bija izpētīt mākoņskaitļošanas pakalpojumu sniedzēju piedāvātās iespējas, kuras ietekmē multireģionālas sistēmas, kā arī izpētīt problēmas, kas rodas multireģionālas, uz tīmekli bāzētas sistēmas izstrādes procesā.

Izvirzītā mērķa sasniegšanai autors izpētīja mākoņskaitļošanas jēdzienu un notikumus, kuri veicināja šīs tehnoloģijas attīstību. Tika aplūkoti mākoņskaitļošanas pakalpojumu līmeņi: SaaS, PaaS un IaaS.

Tā kā viens no multireģionālu sistēmu izstrādes iemesliem ir juridiskie regulējumi, pētījuma otrajā daļā autors pētīja juridiskos ierobežojumus saistībā ar mākoņskaitļošanu, un kā dažādu valstu likumdošanas ietekmē uzņēmumu darbību, kad mākoņskaitļošana tiek izmantota, lai glabātu un apstrādātu personu datus ārpus pilsoņa valsts jurisdikcijas. Tika aplūkoti dažādu valstu likumi, kas attiecas uz datu aizsardzību, kā arī valdības uzraudzību un tiesībām. Tika pētīti arī neseni notikumi, kā ar mākoņskaitļošanu saistīti regulējumi ietekmējuši tādu uzņēmumu darbību kā *Google*, *Twitter*, *LinkedIn* u.c.

Darba trešajā daļā tika salīdzināti 3 populāri mākoņskaitļošanas pakalpojumu sniedzēji - *Amazon Web Services*, *Google Cloud Platform* un *Microsoft Azure*. Pakalpojumu sniedzēju salīdzinājums tika veikts, balstoties uz multireģionālu sistēmu vajadzībām. Tika aplūkoti tādi faktori kā komūnas atbalsts, automātiskas mērogojamības iespējas, pakalpojumu izmaksas u.c. Salīdzinājuma konspektīvam apskatam autors izveidoja tabulu ar katra pakalpojumu sniedzēja priekšrocībām un trūkumiem, kuru iespējams izmantot izstrādājamās sistēmas visatbilstošākā mākoņskaitļošanas pakalpojuma sniedzēja izvēlē.

Pētījuma noslēgumā tika veikta gadījuma izpēte uzņēmumā *Funderful*, kur autors piedalījās multireģionālas sistēmas plānošanā un izstrādē. Tika analizētas problēmas, ar kurām sistēmas uzturēšanas un izstrādes gaitā uzņēmuma komanda sastapās, kā arī tika aprakstīti reāli, produkcijā izmantojami problēmu risinājumi. Atsevišķām problēmām autors izstrādāja arī alternatīvus risinājumus. Sistēmu veiksmīgi izmanto gandrīz 20 augstākās izglītības iestādes visā pasaulē.

## SECINĀJUMI

Mākoņskaitļošanas attīstība sniedz iespējas, kādas nekad iepriekš informācijas tehnoloģiju industrijā nav bijušas. Izmantojot mākoņskaitļošanu, iespējams savas sistēmas izvietot uz infrastruktūras, kas atrodas citā kontinentā, patērējot ievērojami mazāk resursu un neieguldot naudu savas globālas infrastruktūras izveidē.

Mākoņskaitļošanas izmantošanai tīmeklī bāzētu sistēmu uzturēšanā jāņem vērā dažādu valstu regulējumi attiecībā uz personu datu glabāšanu un apstrādi. Tie var būtiski ietekmēt biznesa iespējas un resursu patēriņu sistēmu pārveides vai zaudētas klientūras dēļ.

Mākoņskaitļošanas pakalpojuma sniedzēja izvēle var būtiski ietekmēt multireģionālas sistēmas izstrādes ātrumu un sarežģītību, jo migrācija no viena pakalpojuma sniedzēja uz citu aizņem ievērojamu laika daudzumu. Veicot izvēli, jāņem vērā tādi faktori kā potenciālās izmaksas, automātiskās izvietojšanas iespējas, automātiskas mērogojamības iespējas, pieejamie serveru monitoringa rīki, slodzes līdzsvarošanas iespējas, komūnas atbalsts, tehniskais atbalsts, pakalpojumu sniedzēju datu centru lokācijas, kā arī piedāvātie domēnu nosaukumu pakalpojumi. Liela daļa no pētīto mākoņskaitļošanas pakalpojumu sniedzēju piedāvātās funkcionalitātes ir līdzīga, tāpēc izvēlē jāizvērtē arī sekundārās funkcionalitātes, kur novērojamas daudz lielākas atšķirības.

Apsverot visus salīdzinājumā izmantotos faktoros, pēc autora domām, labākā no apskatītajām pakalpojumu sniedzēju izvēlēm ir *Amazon Web Services*, jo:

- mākoņskaitļošanas pakalpojumu tirgū *AWS* ir vislielākā pieredze;
- pēc veikto eksperimentu rezultātiem tam ir visplašākais komūnas atbalsts;
- *AWS* pieejamās automātiskās mērogojamības iespējas ir labākas nekā aplūkotojāmiem konkurentiem;
- *AWS* piedāvā automātiskās izvietojšanas pakalpojumus ar 2 veidu izvietojšanu un tam ir augstāks *StackShare* vērtējums, salīdzinot ar konkurentiem;
- *AWS* ir industrijā augsti vērtēts domēnu nosaukumu serviss ar pieejamu programmsaskarni.

Pēc veiktās gadījuma izpētes uzņēmuma *Funderful* sistēmai var secināt, ka

- kamēr ir problēmas, kuras sistēmām būs kopīgas, iespējams sastapties ar problēmām, kas būs sistēmai specifiskas un kurām gatavu risinājumu vai risinājumu ideju atrast ir grūti vai pat neiespējami;
- uzturēšanas izmaksas multireģionālai sistēmai visbiežāk būs lielākas nekā analogai standarta sistēmai;

- veidojot jaunu sistēmu, ir vērts apsvērt iespēju izstrādāt jau sākotnējās sistēmas iestrādes, paredzot, ka visus personu datus nebūs iespējams glabāt vienuviet. Lai arī jaunu sistēmu izstrādes sākumā šādi faktori tiek nereti aizmirsti, tas var ilgtermiņā ietaupīt daudz resursu;
- ievērojama daļa *Funderful* sistēmā izmantoto trešo pušu bibliotēku nav izstrādātas, paredzot multireģionālu sistēmu risinājumus, kas vedina secināt, ka arī plašākā apmērā situācija ir līdzīga.

Multireģionālu, globālu sistēmu izstrāde arī nākotnē nekļūs vienkāršāka, personu datu aizsardzības regulējumi dažādās valstīs mainās katru gadu. Autors vedina turpmākus pētījumus veikt saistībā ar populāru trešo pušu bibliotēku iespējām pielāgoties multireģionālu sistēmu tirgum, kā arī mākoņskaitļošanas pakalpojumu sniedzēju darbībām, kas šāda veida sistēmu izstrādi un uzturēšanu padara ērtāku.

## IZMANTOTĀ LITERATŪRA UN AVOTI

- [AAC 17] “AWS Activate”, *Amazon Web Services*, 2017; [tiešsaiste]. [atsauce 11.05.2017.]. Pieejams internetā: <https://aws.amazon.com/activate/>
- [ACD 17] “What Is AWS CodeDeploy?”, *Amazon Web Services*, 2017; [tiešsaiste]. [atsauce 23.05.2017.]. Pieejams internetā: <http://docs.aws.amazon.com/codedeploy/latest/userguide/welcome.html>
- [ACW 17] “Amazon CloudWatch Documentation”, *Amazon Web Services*, 2017; [tiešsaiste]. [atsauce 15.05.2017.]. Pieejams internetā: <https://aws.amazon.com/documentation/cloudwatch/>
- [ADO 17] “Azure DNS overview”, *Microsoft Azure*, 2017; [tiešsaiste]. [atsauce 13.04.2017.]. Pieejams internetā: <https://docs.microsoft.com/hu-hu/azure/dns/dns-overview>
- [AEB 17] “Amazon Elastic Block Store”, *Amazon Web Services*, 2017; [tiešsaiste]. [atsauce 30.04.2017.]. Pieejams internetā: <https://aws.amazon.com/ebs/>
- [AEC 17] “Amazon EC2 Instance Types”, *Amazon Web Services*, 2017; [tiešsaiste]. [atsauce 01.04.2017.]. Pieejams internetā: <https://aws.amazon.com/ec2/instance-types/>
- [AGI 17] “AWS Global Infrastructure”, *Amazon Web Services*, 2017; [tiešsaiste]. [atsauce 11.05.2017.]. Pieejams internetā: <https://aws.amazon.com/about-aws/global-infrastructure/>
- [AGO 17] “Autoscaling Groups of Instances”, *Google Cloud Platform*, 2017; [tiešsaiste]. [atsauce 06.04.2017.]. Pieejams internetā: <https://cloud.google.com/compute/docs/autoscaler/>
- [AMS 17] “Amazon S3”, *Amazon Web Services*, 2017; [tiešsaiste]. [atsauce 07.04.2017.]. Pieejams internetā: <https://aws.amazon.com/s3/>
- [AND 17] Anderson M. “What is Load Balancing?”, *Digital Ocean*, 2017; [tiešsaiste]. [atsauce 16.05.2017.]. Pieejams internetā: <https://www.digitalocean.com/community/tutorials/what-is-load-balancing>
- [ARO 17] “Amazon Route 53”, *Amazon Web Services*, 2017; [tiešsaiste]. [atsauce 13.04.2017.]. Pieejams internetā: <https://aws.amazon.com/route53/>
- [ASP 17] “Auto Scaling Product Details”, *Amazon Web Services*, 2017; [tiešsaiste]. [atsauce 19.04.2017.]. Pieejams internetā: <https://aws.amazon.com/autoscaling/details/>

- [ASU 17] “AWS Support”, *Amazon Web Services*, 2017; [tiešsaiste]. [atsauce 02.04.2017.]. Pieejams internetā: <https://aws.amazon.com/premiumsupport/>
- [ATG 17] “About the Google Cloud Platform Services”, *Google Cloud Platform*, 2017; [tiešsaiste]. [atsauce 13.05.2017.]. Pieejams internetā: <https://cloud.google.com/docs/overview/cloud-platform-services>
- [AUT 17] “Autoscaling”, *Microsoft Azure*, 2017; [tiešsaiste]. [atsauce 05.04.2017.]. Pieejams internetā: <https://docs.microsoft.com/en-us/azure/architecture/best-practices/auto-scaling>
- [AZR 17] “Azure regions”, *Microsoft Azure*, 2017; [tiešsaiste]. [atsauce 11.05.2017.]. Pieejams internetā: <https://azure.microsoft.com/en-us/regions/>
- [AZS 17] “Azure support plans”, *Microsoft Azure*, 2017; [tiešsaiste]. [atsauce 06.04.2017.]. Pieejams internetā: <https://azure.microsoft.com/en-us/support/plans/>
- [BAR 17] Barry D. “Infrastructure as a Service (IaaS)”, *Service Architecture*, 2017; [tiešsaiste]. [atsauce 02.03.2017.]. Pieejams internetā: [http://www.service-architecture.com/articles/cloud-computing/infrastructure\\_as\\_a\\_service\\_iaas.html](http://www.service-architecture.com/articles/cloud-computing/infrastructure_as_a_service_iaas.html)
- [BIP 98] Act, Data Protection. "Basic interpretative provisions." *Office of Public Sector Information*, 1998; [tiešsaiste]. [atsauce 23.05.2017.]. Pieejams internetā: <http://www.legislation.gov.uk/ukpga/1998/29/part/I>
- [BLA 16] Blagovic M. “Google Cloud Deployment Manager”, *V3*, 2016; [tiešsaiste]. [atsauce 01.04.2017.]. Pieejams internetā: <https://impaddo.com/blog/google-cloud-deployment-manager/>
- [CDA 16] “Continuous Deployment to Azure App Service”, *Microsoft Azure*, 2016; [tiešsaiste]. [atsauce 16.05.2017.]. Pieejams internetā: <https://docs.microsoft.com/en-us/azure/app-service-web/app-service-continuous-deployment>
- [CDM 16] “Cloud Deployment Manager”, *Google*; [tiešsaiste]. [atsauce 28.03.2017.]. Pieejams internetā: <https://cloud.google.com/deployment-manager/>
- [CDO 17] “Cloud DNS Overview”, *Google Cloud Platform*, 2017; [tiešsaiste]. [atsauce 14.04.2017.]. Pieejams internetā: <https://cloud.google.com/dns/overview>
- [CLB 17] “Classic Load Balancer Details”, *Amazon Web Services*, 2017; [tiešsaiste]. [atsauce 26.05.2017.]. Pieejams internetā: <https://aws.amazon.com/elasticloadbalancing/classicloadbalancer/>

- [CLL 17] “Cloud locations”, *Google Cloud Platform*, 2017; [tiešsaiste]. [atsauce 11.05.2017.]. Pieejams internetā: <https://cloud.google.com/about/locations/>
- [CPR 17] “Cloud Products”, *Amazon Web Services*, 2017; [tiešsaiste]. [atsauce 11.05.2017.]. Pieejams internetā: <https://aws.amazon.com/products/>
- [CSG 17] “Cloud Storage”, *Google Cloud Platform*, 2017; [tiešsaiste]. [atsauce 07.04.2017.]. Pieejams internetā: <https://cloud.google.com/storage/>
- [CST 17] “Cloud Storage”, *Webopedia*, 2017; [tiešsaiste]. [atsauce 24.05.2017.]. Pieejams internetā: [http://www.webopedia.com/TERM/C/cloud\\_storage.html](http://www.webopedia.com/TERM/C/cloud_storage.html)
- [DER 14] De Rico J. "Cloud computing: personal information and government surveillance" *Lexology*, 2014; [tiešsaiste]. [atsauce 21.05.2017.]. Pieejams internetā: <http://www.lexology.com/library/detail.aspx?g=7f5abc3c-589a-4223-ad68-5dfa3fd7386d>
- [DER 17] “DB-Engines Ranking - Trend of MongoDB Popularity”, *DB-Engines*, 2017; [tiešsaiste]. [atsauce 14.04.2017.]. Pieejams internetā: [https://db-engines.com/en/ranking\\_trend/system/MongoDB](https://db-engines.com/en/ranking_trend/system/MongoDB)
- [DFC 15] "Deadline for Compliance with Russian Localization Law Set for September 1, 2015" *Hunton & Williams*, 2015; [tiešsaiste]. [atsauce 12.04.2017.]. Pieejams internetā: <https://www.huntonprivacyblog.com/2015/01/02/deadline-for-compliance-with-russian-localization-law-set-for-september-1-2015/>
- [FRI 14] Frier S. "LinkedIn Expands in China With Local Website", *Bloomberg Technology*, 2014; [tiešsaiste]. [atsauce 19.03.2017.]. Pieejams internetā: <https://www.bloomberg.com/news/articles/2014-02-25/linkedin-expands-in-china-with-local-website>
- [HAR 14] Harrell J. “The History and Development of Cloud Computing”, *AEROfs*, 2014; [tiešsaiste]. [atsauce 13.03.2017.]. Pieejams internetā: <https://www.aerofs.com/blog/the-history-and-development-of-cloud-computing-md/>
- [HON 16] Hon WK, Millard C, Singh J, Walden I, Crowcroft J. “Policy, legal and regulatory implications of a Europe-only cloud.” *International Journal of Law and Information Technology*, vol. 24, no. 3, 2016, pp. 251-278.
- [IBM 16] “IBM Opens First Cloud Data Center in South Africa”, *IBM*, 2016; [tiešsaiste]. [atsauce 22.03.2017.]. Pieejams internetā: <http://www-03.ibm.com/press/us/en/pressrelease/49305.wss>

- [IMA 17] “Introduction to Microsoft Azure Storage”, *Microsoft Azure*, 2017; [tiešsaiste]. [atsauce 05.04.2017.]. Pieejams internetā: <https://docs.microsoft.com/en-us/azure/storage/storage-introduction>
- [KIM 14] Kim E. “Amazon Web Services Had 44 Price Cuts And You Can Expect More To Come”, *Business Insider*, 2014; [tiešsaiste]. [atsauce 05.05.2017.]. Pieejams internetā: <http://www.businessinsider.com/amazon-web-services-had-44-price-cuts-2014-7>
- [KUR 15] Kurochkin D, Agabalyan M, Ildzhirinova S. "Russia's New Server Localization Law: Implications for Foreign Companies." *World Data Protection Report*, vol. 15, no. 2, 2015; [tiešsaiste]. [atsauce 22.02.2017.]. Pieejams internetā: <https://www.dechert.com/files/Uploads/Documents/Bloomberg - Russia New Server Localization Law - Dechert LLP - February 2015.pdf>
- [MED 16] Medvedev S. “Data protection in Russian Federation: overview”, *Thomson Reuters Practical Law*, 2016; [tiešsaiste]. [atsauce 28.04.2017.]. Pieejams internetā: <https://uk.practicallaw.thomsonreuters.com/2-502-2227>
- [MEL 11] Mell P, Grance T. “The NIST Definition of Cloud Computing”, *National Institute of Standards and Technology – U.S. Department of Commerce*, 2011; [tiešsaiste]. [atsauce 22.05.2017.]. Pieejams internetā: <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
- [MET 16] Metzger M. “Microsoft and US government clash over Ireland-held cloud data”, *SC Magazine UK*, 2016; [tiešsaiste]. [atsauce 12.03.2017.]. Pieejams internetā: <https://www.scmagazineuk.com/microsoft-and-us-government-clash-over-ireland-held-cloud-data/article/531385/>
- [NAG 15] Nagy G. “Operating System Containers vs. Application Containers”, *RisingStack*, 2015; [tiešsaiste]. [atsauce 16.04.2017.]. Pieejams internetā: <https://blog.risingstack.com/operating-system-containers-vs-application-containers/>
- [OOM 17] “Overview of Monitoring in Microsoft Azure”, *Microsoft Azure*, 2017; [tiešsaiste]. [atsauce 15.05.2017.]. Pieejams internetā: <https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview>
- [POT 15] Porter K. “Microsoft versus the Federal Government; Round Three”, *Data Privacy + Security Insider*, 2015; [tiešsaiste]. [atsauce 18.03.2017.]. Pieejams internetā: <https://www.dataprivacyandsecurityinsider.com/2015/04/microsoft-versus-the-federal-government-round-three/>

- [PST 17] “Payments”, *Stripe*, 2017; [tiešsaiste]. [atsauce 03.04.2017.]. Pieejams internetā: <https://stripe.com/us/payments>
- [REP 14] "Regulation (EU) 2016/679 of the European Parliament and of the council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC", *International Journal of Distributed Sensor Networks*, 2014; [tiešsaiste]. [atsauce 19.05.2017.]. Pieejams internetā: <http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679-d1e6226-1-1>
- [RIC 15] Ricky M, Magalhaes M. “Cloud Data Jurisdiction: The provider, The Consumer and Data Sovereignty”, *Cloud Computing Admin*, 2015; [tiešsaiste]. [atsauce 20.04.2017.]. Pieejams internetā: <http://www.cloudcomputingadmin.com/articles-tutorials/compliance-regulations/cloud-data-jurisdiction-provider-consumer-and-data-sovereignty.html>
- [SES 17] “Sendgrid Email Services”, *SendGrid*, 2017; [tiešsaiste]. [atsauce 12.03.2017.]. Pieejams internetā: <https://sendgrid.com/>
- [SGS 16] Sgstechie. “Platform as a Service Pros & Cons”, *SGS Technologies*, 2016; [tiešsaiste]. [atsauce 17.04.2017.]. Pieejams internetā: <http://www.sgstechnologies.net/it-industry-news/platform-service-pros-cons>
- [STA 17] StackShare, Inc. “StackShare”, 2017; [tiešsaiste]. [atsauce 18.05.2017.]. Pieejams internetā: <https://stackshare.io/trending/tools>
- [STM 17] “Stackdriver Monitoring”, *Google Cloud Platform*, 2017; [tiešsaiste]. [atsauce 09.04.2017.]. Pieejams internetā: <https://cloud.google.com/monitoring/>
- [SUN 14] Sun Y, Zhang J, Xiong Y, Zhu G. "Data security and privacy in cloud computing." *International Journal of Distributed Sensor Networks*, 2014; [tiešsaiste]. [atsauce 14.05.2017.]. Pieejams internetā: <http://journals.sagepub.com/doi/full/10.1155/2014/190903h>
- [SUP 17] “Support”, *Google Cloud Platform*, 2017; [tiešsaiste]. [atsauce 02.04.2017.]. Pieejams internetā: <https://cloud.google.com/support/>
- [TIT 20] "The Information Technology Act, 2000" *Ministry of Law, Justice and Company Affairs (Legislative Department)*, 2000; [tiešsaiste]. [atsauce 23.05.2017.]. Pieejams internetā: <http://unpan1.un.org/intradoc/groups/public/documents/apcity/unpan010239.pdf>

- [WEI 16] Weins K. “AWS vs Azure vs Google Cloud Pricing: Compute Instances”, *Right Scale*, 2016; [tiešsaiste]. [atsauce 01.04.2017.]. Pieejams internetā: <http://www.rightscale.com/blog/cloud-cost-analysis/aws-vs-azure-vs-google-cloud-pricing-compute-instances>
- [WIA 17] “What is Azure”, *Microsoft Azure*, 2017; [tiešsaiste]. [atsauce 25.05.2017.]. Pieejams internetā: <https://azure.microsoft.com/en-us/overview/what-is-azure/>
- [WOT 15] Worth D. “Microsoft and US government clash in court over cloud data demands”, *V3*, 2015; [tiešsaiste]. [atsauce 01.04.2017.]. Pieejams internetā: <https://www.v3.co.uk/v3-uk/news/2425489/microsoft-and-us-government-clash-in-court-over-cloud-data-demands>
- [WPC 17] “Worldwide Public Cloud Services Spending Forecast to Reach \$122.5 Billion in 2017, According to IDC”, *IDC*, 2017; [tiešsaiste]. [atsauce 05.04.2017.]. Pieejams internetā: <http://www.idc.com/getdoc.jsp?containerId=prUS42321417>
- [IMA 17] “Introduction to Microsoft Azure Storage”, *Microsoft Azure*, 2017; [tiešsaiste]. [atsauce 05.04.2017.]. Pieejams internetā: <https://docs.microsoft.com/en-us/azure/storage/storage-introduction>

# PIELIKUMI

## 1. pielikums

Risinājuma pirmkods reģionu datu bāžu savienojumu iegūšanai

```
1. # Get current connection configuration
2. current_config = ActiveRecord::Base.connection_config
3.
4. # Get all database configurations
5. all_configurations = Rails.configuration.database_configuration
6.
7. # Get all configurations for Rails.env, except default configuration
8. region_configs = all_configurations.select do |key|
9.   key.include? "#{Rails.env}_" and
10.   current_config.symbolize_keys.to_s !=
11.   all_configurations[key].symbolize_keys.to_s
12. end
13. CLIENT_CONNECTION_POOL = region_configs.map do |name, config|
14.   Object.class_eval <<-EOS
15.     class Client#{name.camelize} < ActiveRecord::Base
16.       establish_connection("#{config}")
17.       self.table_name = :clients
18.     end
19.   EOS
20.
21.   # Makes constant out of string, so it becomes as class reference
22.   "Client#{name.camelize}".constantize
23. end
24.
25.
26. # Made second pool of connections, because in Admin panel will change
27. # self.table_name, which is not thread safe.
28. # Also establish_connection is not threadsafe
29. ADMIN_CONNECTION_POOL = region_configs.map do |name, config|
30.   Object.class_eval <<-EOS
31.     class Admin#{name.camelize} < ActiveRecord::Base
32.       REGION = "#{name.split('_')[1]}"
33.       establish_connection("#{config}")
34.     end
35.   EOS
36.
37.   # Makes constant out of string, so it becomes as class reference
38.   "Admin#{name.camelize}".constantize
39. end
```

**1.1. att. Reģionu datu bāžu savienojumu iegūšana**

## Risinājuma pirmkods reģionu datu sinhronizācijai

```
1. class Client < ActiveRecord::Base
2.   [..]
3.   after_commit :duplicate_change, on: [:create, :update]
4.   after_commit :duplicate_delete, on: [:destroy]
5.
6.   [..]
7.   def duplicate_change
8.     # Create hash with columns updated to update other Database Client models
9.     updated_values = {}
10.    previous_changes.each do |column, value_arr|
11.      # value_arr[0] -> prev value, value_arr[1] -> new value
12.      updated_values[column] = value_arr[1]
13.    end
14.
15.    return if updated_values.blank?
16.
17.    CLIENT_CONNECTION_POOL.each do |foreign_client|
18.      client = foreign_client.where(id: self.id).first_or_initialize
19.      client.update(updated_values)
20.    end
21.  end
22.
23.
24.  def duplicate_delete
25.    CLIENT_CONNECTION_POOL.each do |foreign_client|
26.      foreign_client.where(id: self.id).destroy_all
27.    end
28.  end
29. end
```

**2.1. att. Reģionu datu sinhronizācija**

```
1. Rails.application.routes.draw do
2.   [...]
3.   constraints(WebhookRegion) do
4.     post 'webhook/sendgrid' => 'webhook#sendgrid'
5.     post 'webhook/stripe' => 'webhook#stripe'
6.     [...]
7.   end
8.   [...]
9. end
```

### 3.1. att. Aizķeru galapunkti sistēmā

```
1. class WebhookController < ApplicationController
2.   [...]
3.
4.   def stripe
5.     [...]
6.
7.     case json["type"]
8.     when "invoice.payment_succeeded"
9.       get_person_and_recurring_donation_from_stripe_params(
10.        json_data_object["customer"],
11.        json_data_object["subscription"],
12.        )
13.
14.       # If recurring donations can't be found it must be in another regions
database
15.       unless @recurring_donation.present?
16.         redirect_to_next_region
17.         return render status: 200, json: { status: 'ok' }
18.       end
19.
20.     [...]
21.   end
22.
23.   [...]
24. end
```

### 3.2. att. Stripe aizķeru pieprasījumu apstrāde

```

1. class WebhookController < ApplicationController
2.   [..]
3.
4.   def redirect_to_next_region(options={})
5.     current_region = Rails.application.config.region
6.     if options[:next_region].present?
7.       next_region = options[:next_region]
8.     else
9.       next_region_index =
10.        Rails.application.config.regions.index(Rails.application.config.region) + 1
11.       next_region = Rails.application.config.regions[next_region_index]
12.     end
13.
14.     uri = URI.parse(request.original_url)
15.     url = "#{uri.scheme}://#{uri.host.gsub(current_region,
16.        next_region)}"
17.
18.     if next_region.present?
19.       conn = Faraday.new(:url => url) do |faraday|
20.         faraday.request :url_encoded
21.         faraday.response :logger
22.         faraday.adapter Faraday.default_adapter
23.       end
24.
25.       conn.post do |req|
26.         req.url uri.path
27.         req.headers['Content-Type'] = 'application/json'
28.         req.body = (options[:body] || params).to_json
29.       end
30.     else
31.       [..]
32.     end
33.   end
34. end

```

### 3.3. att. Aizķeru novirzīšana uz atbilstošo reģionu

```

1. namespace :backup do
2.   desc 'Perform backup before deployment'
3.   task :perform_backup do
4.     regions = []
5.     on roles(:primary_db) do |server|
6.       region = server.instance_variable_get('@properties').region
7.
8.       if !regions.include? region
9.         regions << region
10.        if test("[ -f #{release_path}/config/backup.rb ]")
11.          execute("~/rbenv/bin/rbenv exec backup perform --trigger before_deployment --
config-file # {release_path}/config/backup.rb --tmp-path #{release_path}/tmp")
12.        end
13.      end
14.    endā
15.  end
16. end

```

#### 4.1. att. Aizķeru novirzīšana uz atbilstošo reģionu

```

1. namespace :database do
2.   desc "Create custom database.yml configurations"
3.   task "generate_database_yml" do
4.     on roles(:app) do |server|
5.       # Create database yml file with the specific region
6.       region = server.instance_variable_get('@properties').region
7.       location = fetch(:template_dir, "config/deploy/shared") + '/database.yml.erb'
8.       template = File.read(location)
9.       config = ERB.new(template)
10.      b = binding
11.      b.local_variable_set(:region, region)
12.      [..]
13.
14.      # Create necessary folders and upload files
15.      execute "mkdir -p #{shared_path}/db"
16.      execute "mkdir -p #{shared_path}/config"
17.      upload! File.open(database_yml), "#{shared_path}/config/database.yml"
18.      upload! File.open(config_file), "/home/ubuntu/my.conf"
19.      [..]
20.    end
21.  end
22. end

```

#### 4.2. att. Datu bāzu savienojumu iestatījumu datņu izveide uz AWS virtuālajām mašīnām

```
1. production_us: &production_us
2.   database: funderful_production
3.   username: *****
4.   password: *****
5.   adapter: mysql2
6.   host: funderful-production.*****.*****.amazonaws.com
7.   port: ****
8.   encoding: utf8mb4
9.
10. production_eu: &production_eu
11.   database: funderful_production
12.   username: *****
13.   password: *****
14.   adapter: mysql2
15.   host: funderful-production.*****.*****.amazonaws.com
16.   port: ****
17.   encoding: utf8mb4
18.
19. production:
20.   <<: *production_<%= region %>
```

#### **4.3. att. Datu bāžu savienojumu iestatījumu datnes veidne**