Tampereen ammattikorkeakoulu

Ohjelmistorajapinnan toteutus Azure Functions -pilvipalvelussa

Ari Suvanto

OPINNÄYTETYÖ
Toukokuu 2022

Tieto- ja viestintätekniikka
Ohjelmistotekniikan koulutusohjelma

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tieto -ja viestintätekniikka
Ohjelmistotekniikan koulutusohjelma

SUVANTO, ARI:
Ohjelmistorajapinnan toteutus Azure Functions -pilvipalvelussa

Opinnäytetyö 27 sivua
Toukokuu 2022

Azure Functions on yksi Microsoft Azure pilvipalvelualustan tarjoamista palveluista. Tämän lisäksi Azure tarjoaa kattavan valikoiman muita erilaisia palveluja tämän työn ja muiden sovellusten tueksi. Monitorointi- ja valvontatyökalut ovat esimerkkejä näistä. Azure Functions ei kuitenkaan ole sidottuna niiden käyttöön, joten aloitteleva ohjelmoijakin saa pystytettyä yksinkertaisen funktiosovelluksen.

Työn päämääränä on kehittää API-rajapintoja Azure Function -palvelua käyttäen. Rajapinnat mallinnetaan olemassa olevan ASP.NET MVC -sovelluksen PTMaintenancen pohjalta. PTMaintenance on tarkoitettu yksittäisten asennusten luontiin ja käyttäjätilien ylläpitoon hajautetussa järjestelmässä. Toisena esimerkkinä tehdään myös html-to-pdf -sovellus, joka toimii Azure Functions -ympäristössä. Html-to-pdf -sovellus muuntaa html-muotoisia harjoiteohjeita pdf-tiedostoiksi.

Sovellusten kehitystyö tapahtuu sekä Azure-portaalissa, että Visual Studio Code -kehitysympäristössä. Portaalista nähdään sovelluksen käyttöön liittyvää monitorointidataa ja säädetään muita lisäpalveluita, kuten tallennustilin ja skaalaustyökalujen käyttöä. Visual Studio Codessa puolestaan toteutetaan itse ohjelmointityö ja paikallinen testaus.

Opinnäytetyössä tutkitaan Azuren toiminnallisuuksia ja vaihtoehtoisia pilvipalvelualustoja työn toteuttamiseksi. Aluksi esitellään yleistietoa Azuresta ja Azure-portaalin käytöstä. Sitten käydään läpi API-rajapintojen toteutus ja sen lisäksi vaihtoehtoisia toteutustapoja, jotka voivat olla myös lisäominaisuuksia sovellukselle tulevaisuudessa. Lopuksi käydään läpi toteutussuunnitelmaa ja tehdään työstä johtopäätökset.

Asiasanat: azure, asp.net, api-rajapinta, pilvipalvelut, palvelimeton

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Information Technology
Computer Systems Engineering

SUVANTO, ARI:
Creating API in Azure Functions cloud service

Bachelor's thesis 27 pages
May 2022

Azure Functions is one of many cloud services provided by Microsoft Azure. Azure offers a large selection of different services for Azure Functions for this work and for other services. Monitoring tools are one of these services. However, Azure Functions is not tied to their use so even an amateur programmer can create a simple function application without them.

The aim of the work is to create APIs using Azure Functions program. The interfaces will be modeled after existing ASP.NET MVC application PTMaintenance. PTMaintenance is used to create single trial accounts and to upkeep user accounts in distributed systems. There will be also created html-to-pdf application which will work in Azure Functions environment. Html-to-pdf application will convert html-formatted exercise instructions to pdf-files.

Programming development happens both in Azure portal and in Visual Studio Code. From the portal monitoring data can be observed and other extras such as storage account and scaling tools can be adjusted. Programming and local testing is done in Visual Studio Code environment.

The thesis studies Azure’s functionalities and alternative cloud programming services. General information about Azure is introduced first and then usage of Azure portal. Then API implementation is introduced along with alternative methods which can also be extras for the program itself in the future. In the end implementation plan and conclusions about the work is shown.

Key words: azure, asp.net, api, cloud services, serverless

SISÄLLYS

LYHENTEET JA TERMIT

MVC	Model-view-controller. Suunnittelumalli, jossa ohjelmiston rakenne on jaettu käyttöliittymään (view), dataan (model) ja ohjelmistologiikkaan (controller).
REST	Representational State Transfer. Arkkitehtuurityyli web-palvelujen luomiseen
API	Application Programming Interface. Rajapinta, jossa ohjelmistot kommunikoivat keskenään.
.Net	Ohjelmistokehyksen kirjasto
On-premise	Ohjelmistopalvelu asiakkaan serverillä ja infrastruktuurissa
Low-code	Ohjelmistokehitystapa, jossa suurin osa toiminnoista toteutetaan kehitysympäristön graafisilla työkaluilla ja tarvitaan vain minimaalinen määrä käsin kirjoitettua koodia.
No-code	Ohjelmistokehitystapa, jossa ei käytetä käsin kirjoitettua koodia lainkaan, vaan kaikki toiminnot määritellään kehitysympäristön graafisilla työkaluilla.
IaaS	Infrastructure-as-a-Service. Palvelinympäristön ulkoistaminen palveluntarjoajalle. Sovellus- ja tietokantapalvelimet, verkkoyhteydet, tietoturvaratkaisut ja niiden ylläpito hankitaan kokonaispalveluna
PaaS	Platforms-as-a-Service. Palvelualustan ulkoistaminen. Sisältää ohjelmistokomponenttien asennukseen, valvontaan ja skaalaukseen tarjottavat työkalut
SaaS	Software-as-a-Service. Tarjotaan sovellus kokonaisuudessaan pilvipalveluna
FaaS	Function-as-a-Service. Palvelittomat (serverless) ohjelmistoratkaisut, tyypillisesti API-rajapintoja
Azure	Microsoftin pilvipalveluympäristö
AWS	Amazon Web Services -pilvipalvelu

1 JOHDANTO

Toimeksiantaja Physiotools Oy:llä on käytössään pitkältä ajalta hyvin erilaisilla teknologiolla tehtyjä ohjelmistorajapintoja. Tavoitteena on yhdenmukaistaa nämä rajapinnat nykyisin paljon käytössä olevan REST API -rajapintarakenteen mukaiseksi ja hyödyntää Azure-pilvipalvelua niiden toteutuksessa.

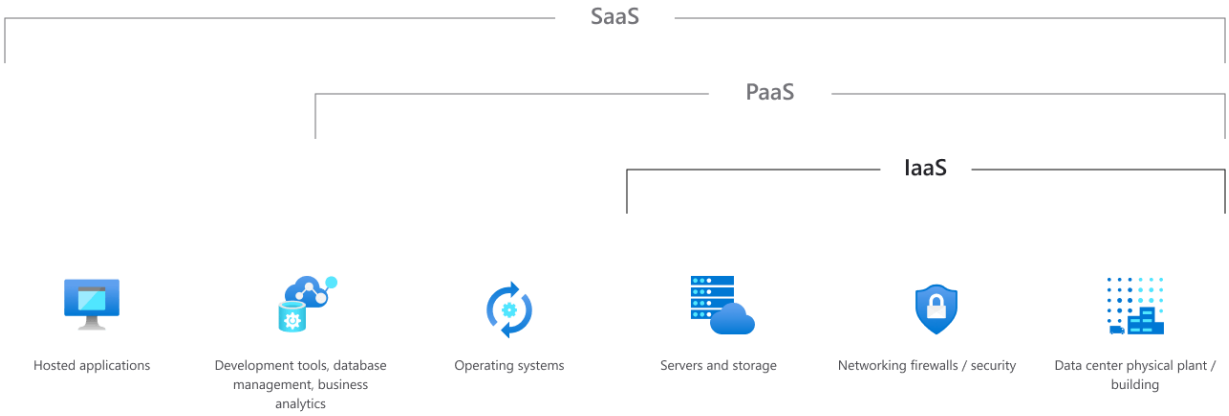
Tarkoituksena on aluksi toteuttaa rajapinnat On-premise -versioina toimeksiantajan omille palvelimille ja sen jälkeen toteuttaa ne Azure-pilvipalvelussa serverless-versioina. Tähän työhön valittiin toteutettavaksi API-rajapinnat, joiden avulla tehdään uusia asennuksia Physiotools-palvelimelle ja muunnetaan html-muotoisia harjoiteohjeita pdf-muotoon.

Toimeksiantaja halusi, että tutkitaan Azure Functions -palvelun sopivuutta ylläkuvatun toimeksiannon toteuttamiseen.

2 MICROSOFT AZURE -PILVIPALVELU

2.1 Yleistä Azuresta

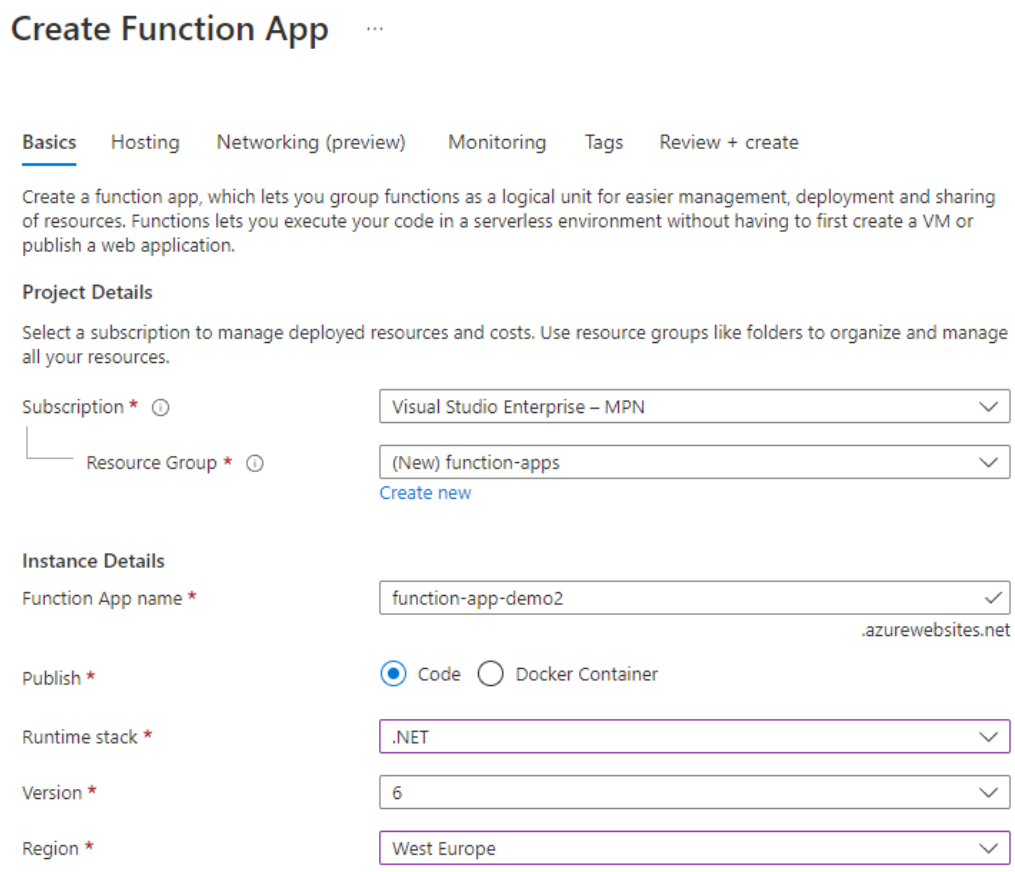
Azure on Microsoftin tarjoama pilvipalvelu, missä on laaja valikoima eri työkaluja ja palveluita. Alimman tason palvelut eli IaaS-palvelut Azuressa tarjoavat virtuaalikoneita, tallennuspalveluita ja tietoturvapalveluita. PaaS-palveluita tarjotaan kehitystyökaluina, tietokantahallintaa ja käyttöjärjestelminä. Loppukäyttäjille tarjotut ohjelmistot ja järjestelmien väliset rajapintapalvelut ovat SaaS -palveluita.



KUVIO 1. Azure-pilvipalvelun IaaS-, PaaS- ja SaaS-palvelukerrokset. (Azure Microsoft, What is Iaas)

2.2 Azure Portal -käyttöliittymä

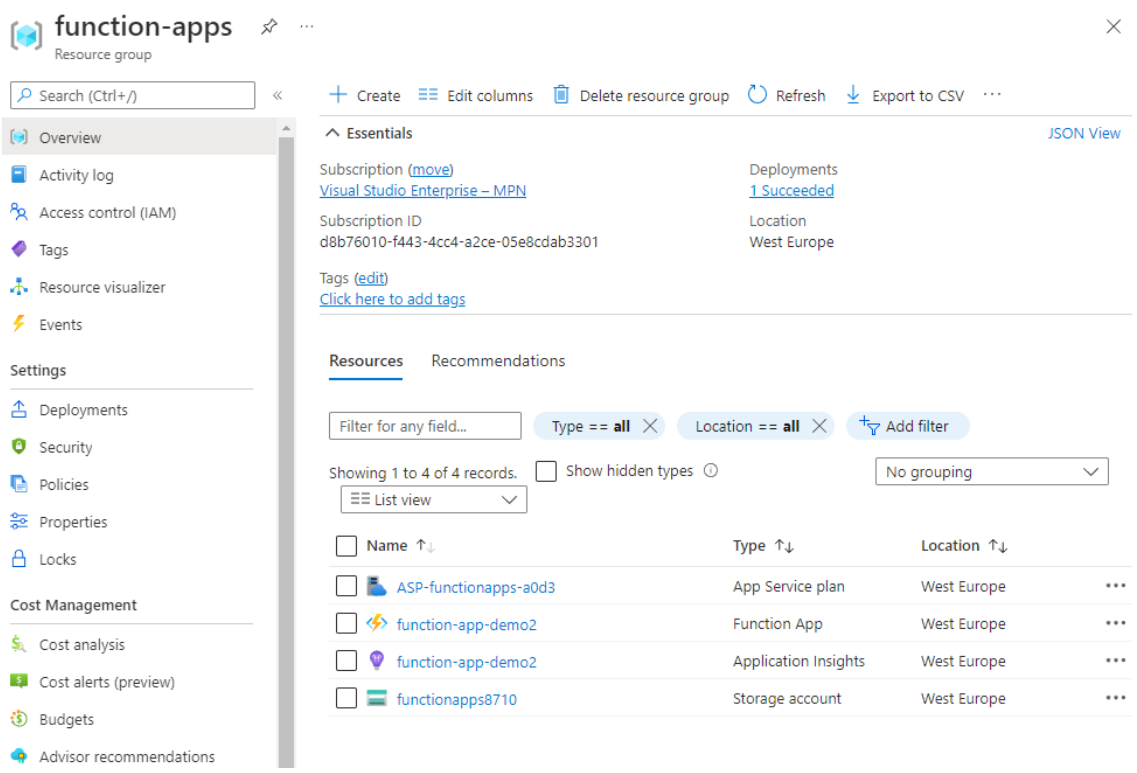
Azure Portal on graafinen web-käyttöliittymä Azuren tarjoamien palveluiden käyttöönottoon, ylläpitoon ja seurantaan. Azureen kuuluu myös erilaisten hinnoittelumallien valitseminen käytön mukaan eli pay-as-you-go, kiinteä hinnoittelu tai kutsujen mukainen hinnoittelu. Azure-portaalista saa säädettyä resurssien skaalautuvuutta lisäämällä suoritintehoa (scale up) tai monistamalla sitä (scale out).



KUVIO 2. Azure-portaalissa funktion määrittely

Kuten kuviosta 2 nähdään, Azure-portaalissa funktiota luodessa sille valitaan oma resurssiryhmä, mihin kaikki luodut resurssit lisätään. Julkaisutapana toimii myös Docker -sovelluksen käyttö ja alueeksi sopii myös esimerkiksi North Europe, jonka sijainti on Irlanti. Lähin konesali on opinnäytetyön kirjoittamisen hetkellä West Europessa Alankomailla.

Hosting-osuudessa valitaan tallennustili ja kulutussuunnitelma, sekä päätetään Linux- tai Windows-pohjaisen käyttöjärjestelmän väliltä. Monitoring-osuudessa puolestaan päätetään Application Insightin käytöstä, millä on oltava sama alue kuin kuvion 2 Basics-osuudessa.

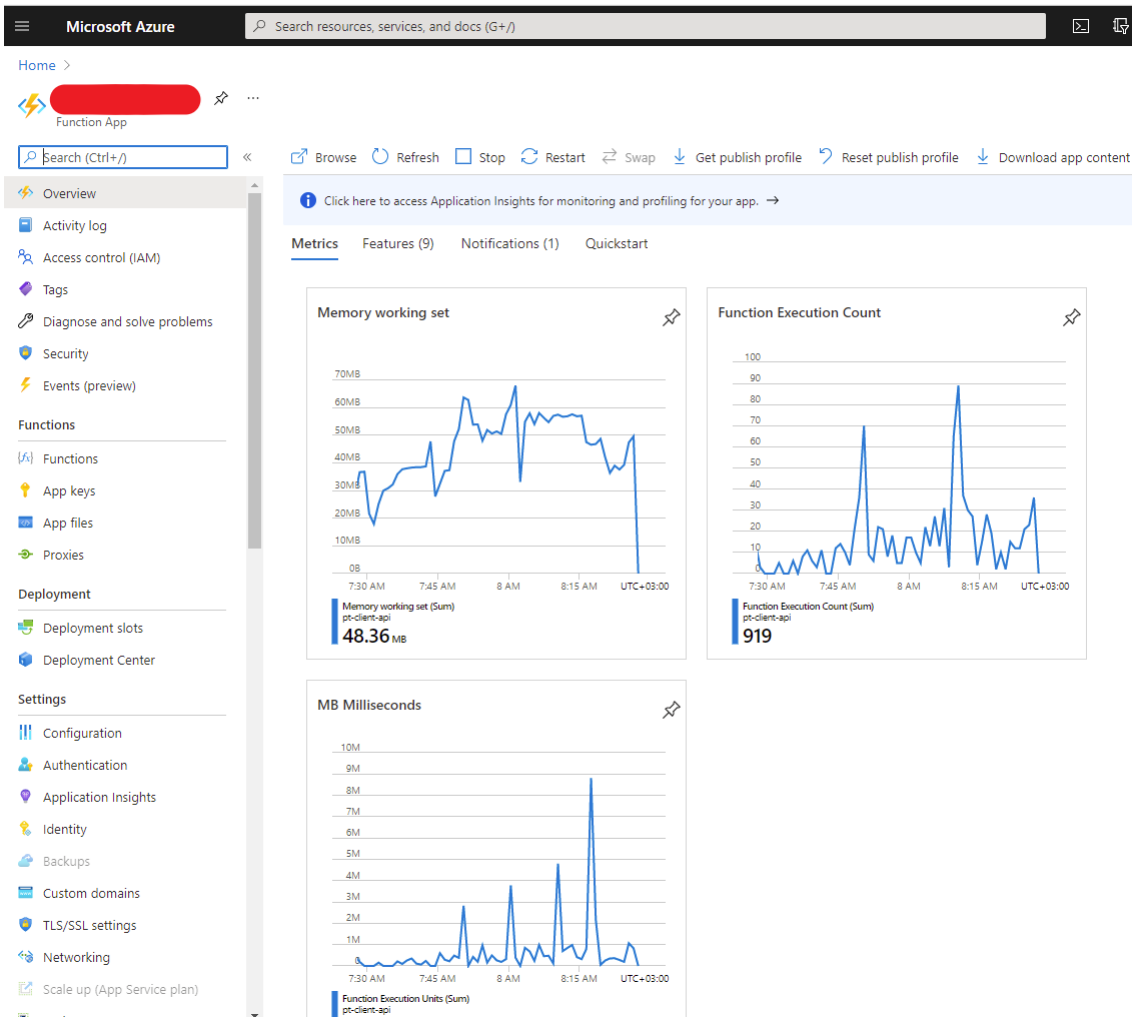


KUVIO 3. Azuressa luotu function-apps -resurssiryhmä.

Kuviosta 3 nähdään luotu resurssiryhmä ja siihen kuuluvat tallennustili, Application Insights, itse funktio ja App Service -suunnitelma. Tähän resurssiryhmään voidaan halutessa lisätä lisää funktioita tai monitoroida olemassa olevia.

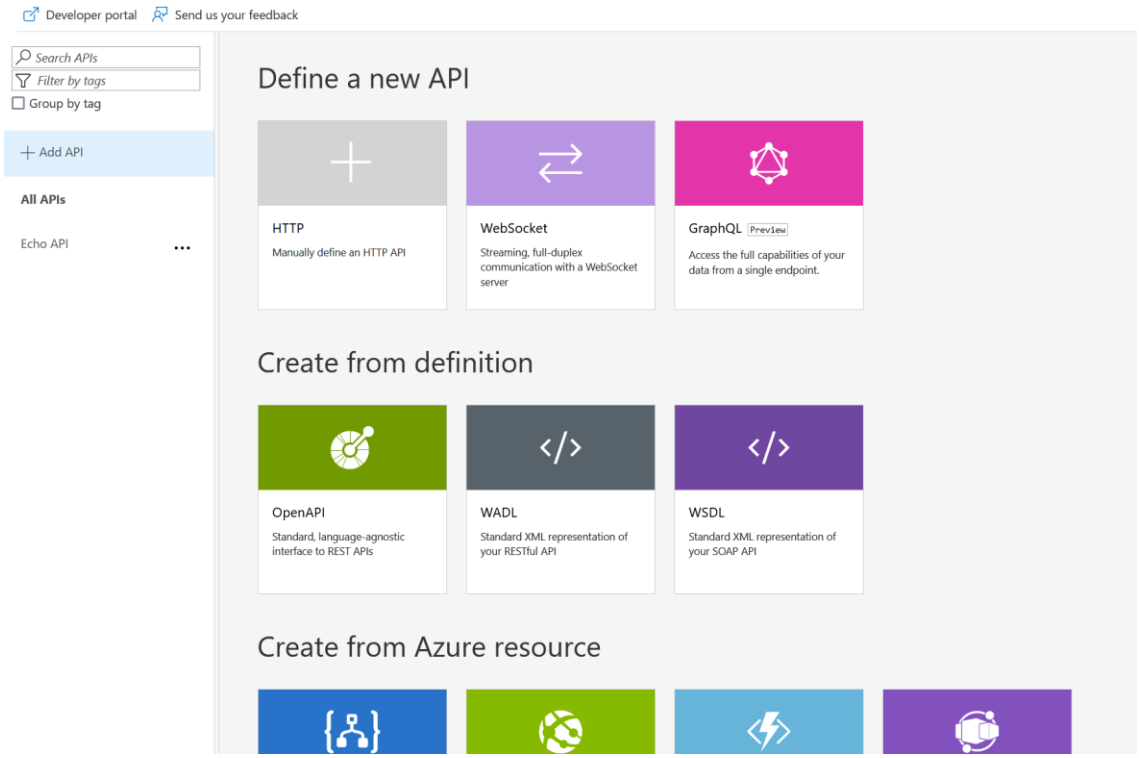
Application Insights auttaa kokenutta käyttäjää automaattisesti löytämään suorituskykyyn vaikuttavia asioita, näyttämään käyttäjien käyttötietoja ja nostamaan sovellustehokkuutta ja käytettävyyttä.

App Service Plan auttaa käyttäjää selvittämään sovelluksen ajamiseen ja skaalautuvuuteen liittyviä tietoja. Sen hinta riippuu valitusta maksutasosta. Shared-maksutasossa maksetaan CPU:n käytöstä minuutteina, Isolated-maksutasossa sovellusta ajavat työkalut määrittävät hinnan ja muilla maksutasoilla virtuaalikoneiden määrä on hintaan vaikuttava tekijä.



KUVIO 4. Funktion monitorointi Azure-ympäristössä.

Kuviosta 4 nähdään Azureen tullutta funktion monitorointidataa, mistä selviää muistinkäyttö, kutsujen käyttömäärä ja datan siirtomäärä. Azuren monitorityökaluilla voidaan myös reaaliaikaisesti reagoida poikkeamiin lähettämällä esimerkiksi viestejä ylläpidolle tai automaattisesti suorittaa korjaustoimenpiteitä.



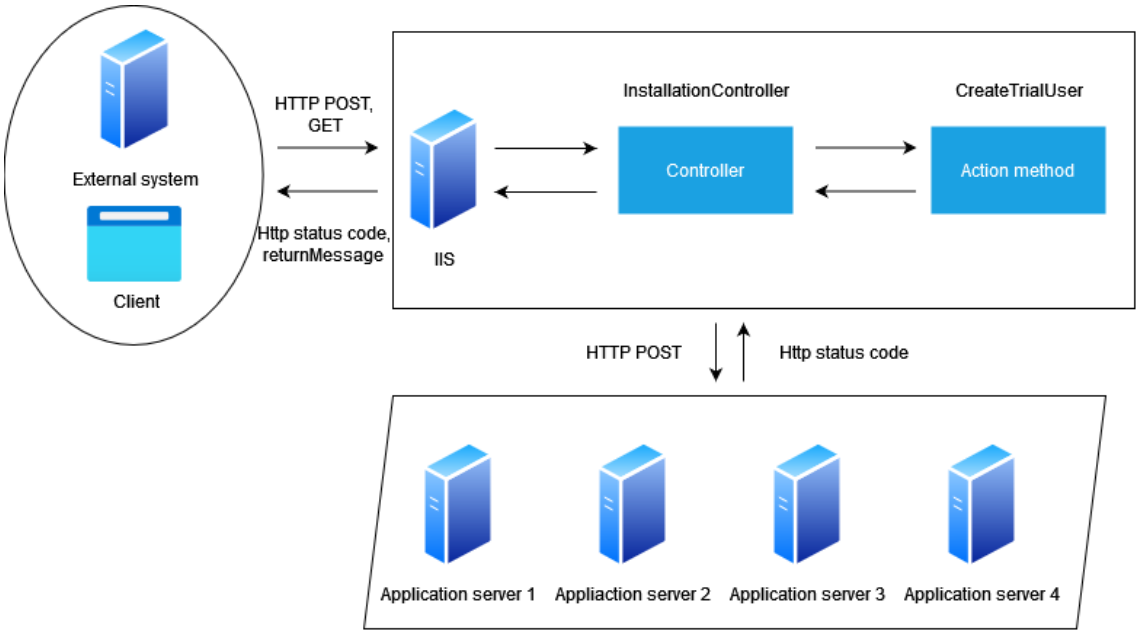
KUVIO 5. API management -työkalu

API managementin avulla voidaan julkaista API-rajapintoja ja hallinnoida API-avaimia ja autentikointia, sekä rajoittaa ja monitoroida API-rajapinnan käyttöä. Julkaisutyökaluissa on välineet rajapinnan dokumentointiin (OAS/Swagger) ja versiointiin. API managementissa kehittäjälle tarjotaan Developer Portal -työkalu, jossa kehittäjä voi tutustua dokumentaatioihin ja esimerkkeihin, ladata SDK:ta ja hallinnoida omia API-avaimia.

3 TEKNOLOGIAT RAJAPINNAN TOTEUTUKSEEN

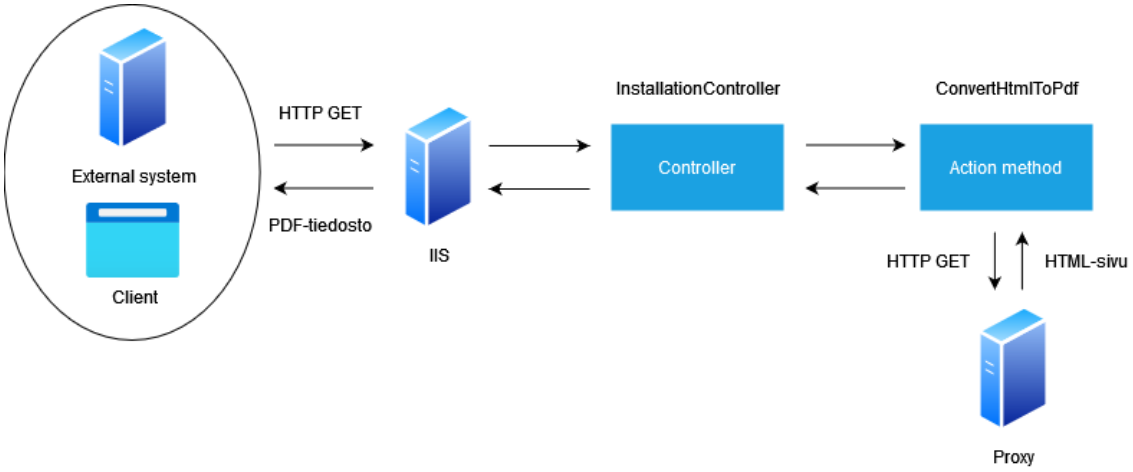
3.1 On-premise ASP.NET Web API -versio

Osana työn toteutusta tehtiin jo käytössä olevalle palvelimelle uusi metodi API-rajapintaan. Yhteistyökumppanin ylläpitämä järjestelmä käyttää rajapinnan metodia lisäämään koekäyttäjiä Physiotoolsin SaaS-palveluun. API-rajapintaa kutsuttaessa lähetetään käyttäjän pakolliset ja valinnaiset parametrit kontrolleriin http-pyynnöillä. Kontrolleri ohjaa viestintäpyynnöt http-pyyntömetodien avulla oikealle action-metodille. Action-metodissa valitaan käyttäjälle lähin palvelin country-parametrin mukaan, kutsutaan palvelimella olevaa API-rajapintaa ja palautetaan http-vastauskoodit kirjautumistietojen kanssa.



KUVIO 6. On-premise ASP.NET Web API -rajapinta koekäyttäjien luontiin.

Kyseisessä systeemissä on kuvion 6 mukainen olemassa oleva järjestelmä, jossa IIS-palvelin on Physiotoolsin virtuaalipalvelimella, jota ylläpidetään itse. Kuvan ulkopuolisia systeemejä voi olla esimerkiksi yhteistyökumppanin järjestelmä, josta halutaan tarjota mahdollisuus luoda Physiotoolsin koekäyttäjiä automaattisesti. Physiotoolsin sovelluspalvelimet ovat sijoitettu ympäri maailmaa palvelemaan kunkin alueen maita.

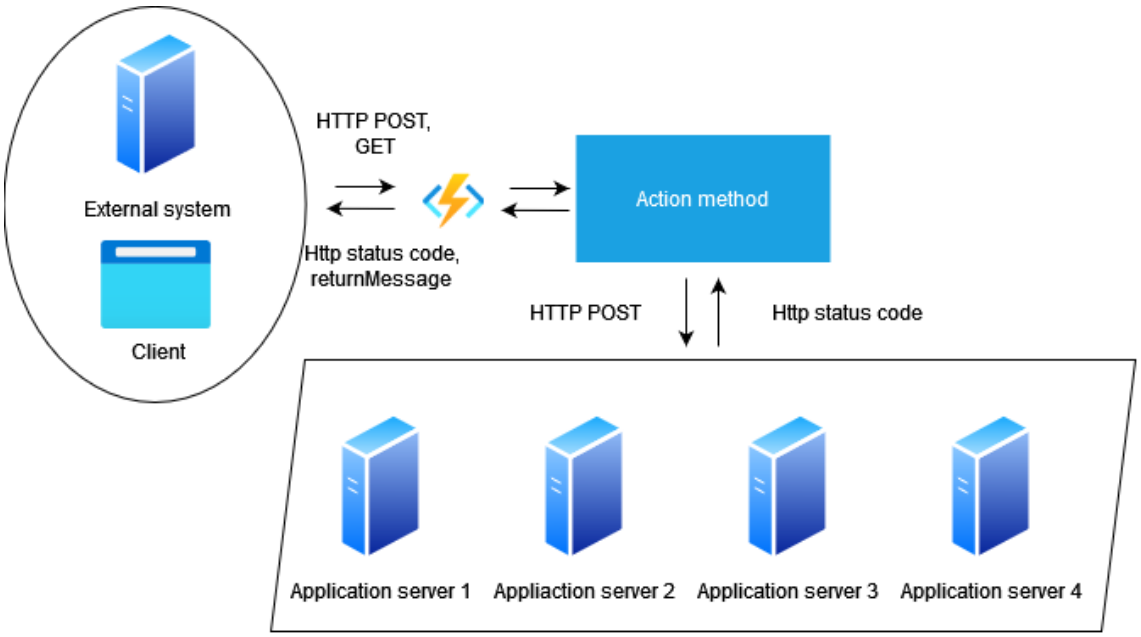


KUVIO 7. On-premise ASP.NET Web API -rajapinta pdf-tiedostojen luontiin.

Html-to-pdf -toteutuksessa (kuvio 7) ohjataan HTTP GET -pyyntö ConvertHtmlToPdf-metodille, joka muuntaa välityspalvelimelta saadun HTML-sivun PDF-tiedostoksi ja palauttaa sen vastauskoodin kanssa kutsuvalle järjestelmälle. Toteutuksessa käytettävät PDF-tiedostot ovat harjoiteohjeita, joissa on piirros- tai valokuvia, ohjetekstejä ja linkkejä ulkopuolisiin videoharjoitteisiin.

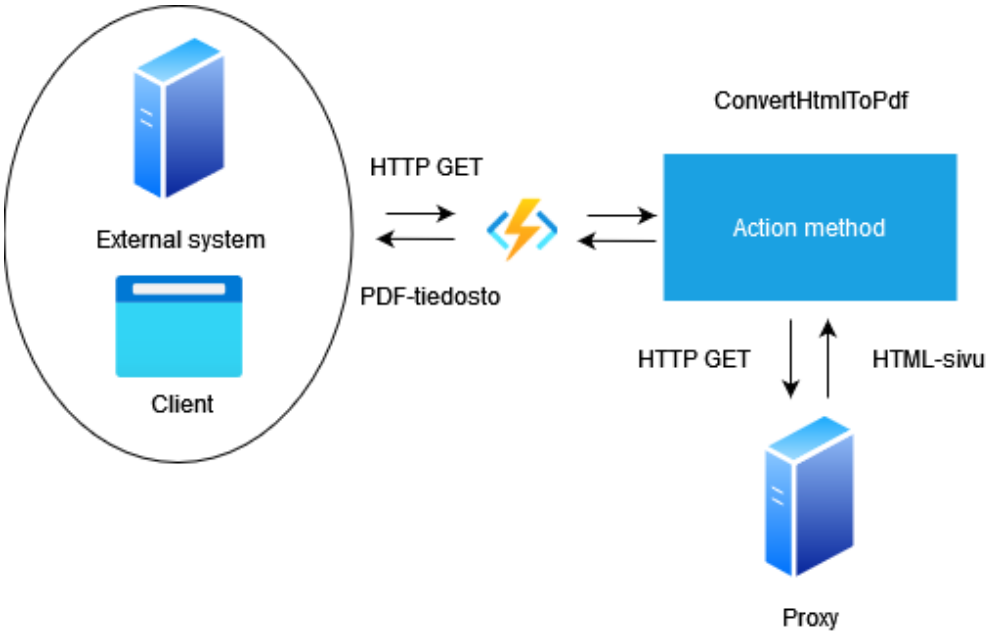
3.2 Azure Function App -toteutukset

Azure Function App on event-pohjainen palvelimeton laskenta-alusta, mikä mahdollistaa rajapintojen ajamisen, debuggaamisen ja monitoroinnin pilvessä ilman serverin tuomaa infrastruktuuria. Funktio itsessään kommunikoi http-triggereiden avulla, jotka palauttavat http-vastauskoodin paluuarvoina. Azure-portaalissa on työkalut funktioiden käytön rajoittamiseen ja monitorointiin. Funktioiden käytön hinnoittelua on mahdollista säätää kuhunkin tapaukseen sopivalla tavalla.



KUVIO 8. Azure Functions -versio Web API-rajapinnasta koekäyttäjien luontiin.

Kuviossa 8 on Azuren kautta toimivan toteutuksen suunnitelma, missä rajapinta sijaitsee Azuressa, mutta toimii samalla tavalla kuin kuvion 6 on-premise -versio. Azure Functions mahdollistaa kutsujen tekemisen ilman on-premise -version IIS-palvelinta ja MVC-kontrolleria. Toteutuksessa ei tarvitse tehdä on-premise -version mukaista kontrolleria ollenkaan, vaan funktio konfiguroidaan vastamaan haluttuun http-pyyntöön Azure-portaalissa.



KUVIO 9. Azure Functions -versio html-tiedostojen luontirajapinnasta.

Html-to-pdf -toteutus (kuvio 9) vastaa on-premise -version (kuvio 7) kutsua, mutta IIS-palvelimen ja kontrollerin on korvannut Azure Functions -palvelu.

3.3 Muita vaihtoehtoisia toteutustapoja

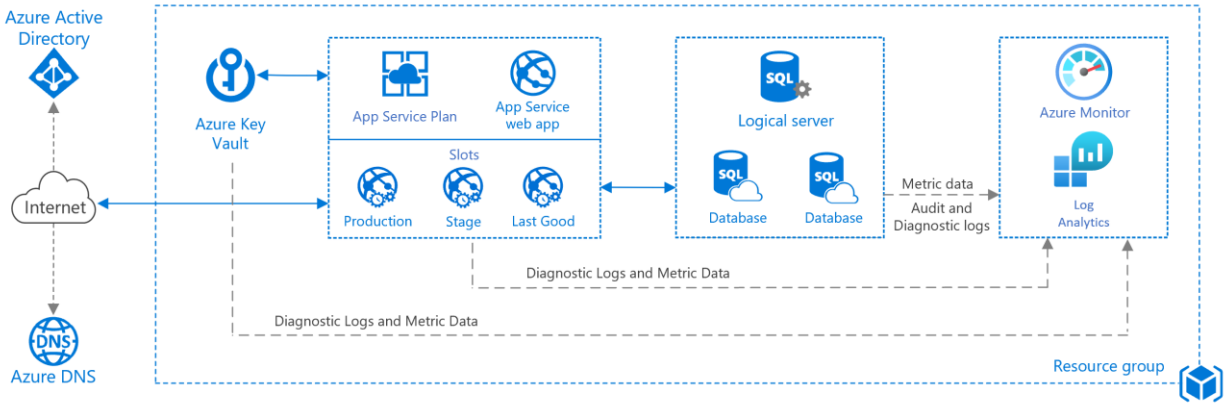
Sama toiminnallisuus olisi mahdollista toteuttaa Azuressa myös muutamilla eri tavoilla. Seuraavaksi on listattu eri vaihtoehtoja ja mietitty niiden eroja Function App -toteutukseen. Itse funktio ei välttämättä tarvitse mitään muutoksia, mutta sen suoritusta ohjataan erilaisten Azuren tarjoamien palveluiden avulla.

3.3.1 Virtuaalikone Azuressa

Yhtenä vaihtoehtona olisi siirtää olemassa oleva toteutus sellaisenaan Azuressa virtuaalikoneelle. Toteutuksesta ei olisi kauheasti hyötyä, koska siinä ei pystyisi hyödyntämään Azuren tuomia lisäominaisuuksia ja hinnoittelu ei olisi edullisempaa kuin nykyisessä järjestelmässä.

3.3.2 App Service -toteutus

Vaihtoehtoina Function App -toteutuksille on siirtää olemassa oleva Web API -toteutus pienin muutoksin App Service -palveluun. Tämä tuo rajapintaan skaalautuvuutta, sekä palvelimen ylläpidosta ei tarvitsisi huolehtia. Hinnoittelu App Service -toteutuksissa tapahtuu esimerkiksi pay-as-you-go -tyylisesti käytön mukaan, jolloin saadaan säästöjä. Koska kuitenkaan kaikkia nykyisiä on-premise -toteutuksessa olevia API-rajapintakutsuja ei ole tarkoituksenmukaista siirtää, niin on yksinkertaisempaa toteuttaa siirrettävät metodit Azure-funktioiden avulla.

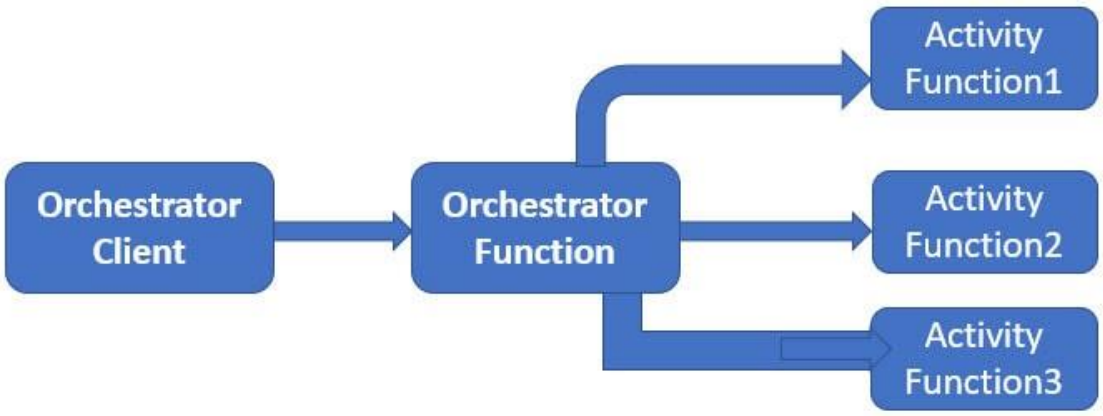


KUVIO 10. Web API -arkkitehtuurikuva. (Microsoft, Basic Web Application)

App Service -palvelussa on mahdollista käyttää monia Azure-portaalin tarjoamia muita palveluja. Esimerkiksi Azure Key Vaultissa voidaan säilyttää salattuja API-avaimia tai Azure Active Directoria voidaan käyttää käyttäjän tunnistamiseen. Azure Monitor tarjoaa valvonta- ja analysointityökalut, mistä nähdään esimerkiksi datan siirtomääriä ja sovelluksen käytön jakautumista eri vuorokaudenaikoihin. Monitoroinnin perusteella voidaan määritellä sääntöjä automaattiselle skaalautumiselle. Datan talletukseen Azure-ympäristö tarjoaa hyvin erilaisia vaihtoehtoja pienillä ohjelmistomuutoksilla, kuten Blob Storage, Cosmos DB tai perinteinen SQL-tietokanta.

3.3.3 Durable Functions -toteutus

Durable Functions on Azure Functions -palvelun laajennus, jonka avulla voidaan kirjoittaa tilallisia funktioita palvelimettomassa ympäristössä. Durable-funktioissa on orkestroinnin käyttöliittymä, joka toimii http-triggerien tapaisesti, sekä orkestrointifunktio, jossa ohjataan eri aktiviteettien eli yksittäisten funktioiden käyttöä. Durable-funktioiden käytöstä löytyy erilaisia funktiokäyttöjen mallinnuksia kuten funktioketjutusmalli ja fan out/fan in -mallinnus. Durable Functions sopii sellaisiin toteutuksiin, joissa yksittäiset toimenpiteet ovat pitkäkestoisia tai toimenpiteet pitää ketjuttaa tai ajaa rinnakkain. Koska tämän toimeksiannon sovelluksessa toimenpiteet ovat suhteellisen lyhytkestoisia, ei Function App -toteutuksen aikaraja (5-10 minuuttia) tule rajoittavaksi.

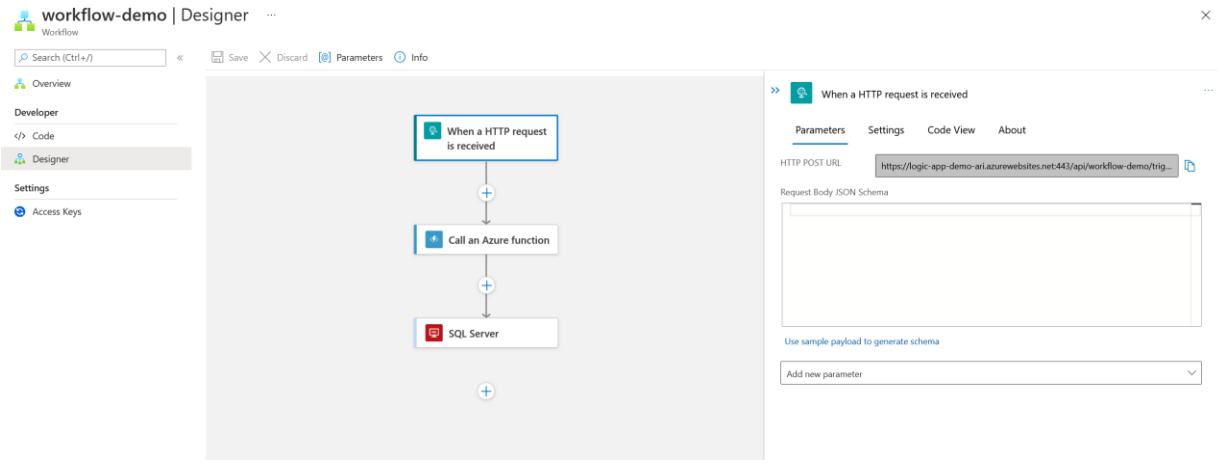


KUVIO 11. Durable Functions -mallinnus (Bijay Kumar 2021.)

Durable-funktioiden toteutukseen voidaan sen tarpeen mukaan käyttää erilaisia sovellusmalleja. Tyypillisesti tällaisia malleja ovat funktioketjutus, fan-out/fan-in eli rinnakkainen suoritus, asynkroninen http API -rajapinta, monitorointi ja käyttäjä interaktio.

3.3.4 Azure Logic Apps -toteutus

Perinteisen ohjelmoinnin lisäksi Azuressa voi toteuttaa low-code- ja no-code -ohjelmointia, mikäli on jo olemassa valmiita toteutuksia, josta tarvittavan palvelun voi rakentaa. Yksittäiset toimenpiteet voivat olla itse tehtyjä Azure-funktioita tai ympäristön tarjoamia valmiita komponentteja tai järjestelmän ulkopuolisten palvelujen kutsujia.

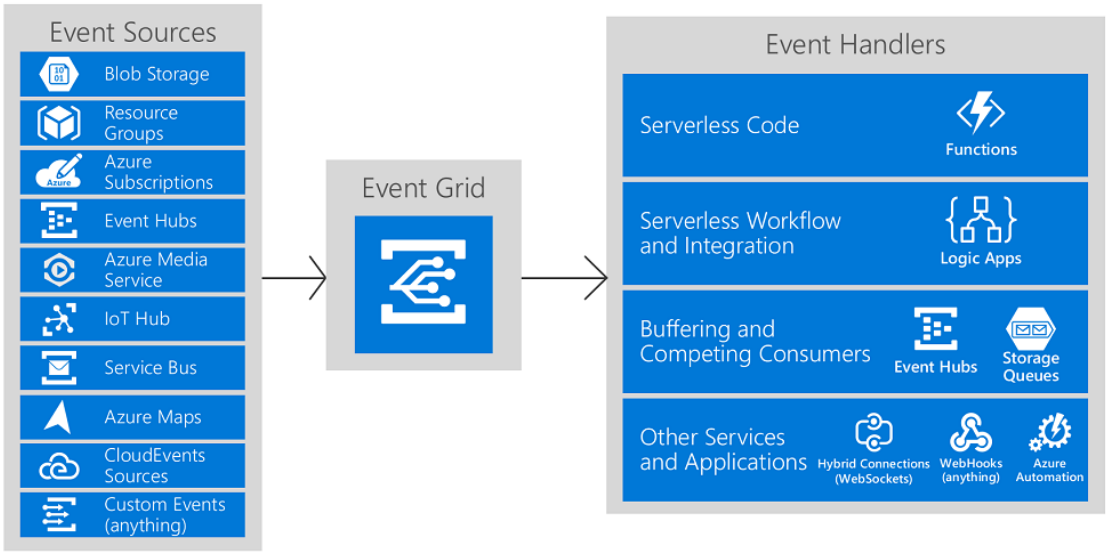


KUVIO 12. Azure Logic Apps Designer -suunnittelutyökalu.

Azure Logic Apps Designer -suunnittelutyökalu on alusta, missä luodaan palveluun työnkulkuja. Jokainen työnkulku sisältää oman triggerin, minkä jälkeen lisätään yksi tai useampi action-metodi.

3.3.5 Azure Event Grid -toteutus

Azure Event Gridin avulla voidaan rakentaa event-pohjaiseen arkkitehtuuriin palvelimettomia sovelluksia. Azure Event Grid -toteutus Function Appin avulla toimisi samalla tavalla kuin kuviossa 8 on esitelty, mutta toteutuksen ulkopuoliset systeemit olisivat monimuotoisempia. Funktion käynnistävä tapahtuma ja sen käsittelijä voidaan konfiguroida hyvin monipuolisesti.



KUVIO 13. Azure Event Grid -mallinnus. (Microsoft, React to Azure Maps events by using Event Grid)

Kuten kuviosta 13 nähdään Azure Event Gridin tapahtumalähteinä ja tapahtumakäsittelijöinä voi toimia monia eri Azuren tarjoamia palveluja. Event Grid pystytään konfiguroimaan niin, että epäonnistunut viesti voidaan lähettää uudestaan. Käyttöoikeuksia voidaan rajata monipuolisesti tarpeen mukaan.

3.3.6 Azure Function valintaperusteet

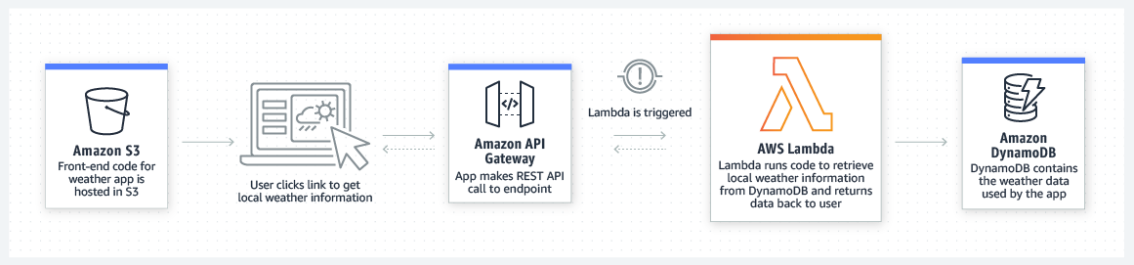
Azure tarjoaa perinteisen hinnoittelun lisäksi komentopohjaista hinnoittelumallia, missä laskutus tapahtuu joko gigatavu per sekunti tai toteutettujen kutsujen määrän mukaisesti. Tämän lisäksi Azuren portaalista löytyy tehokkaita valvonta- ja ylläpitotyökaluja.

Kustannustehokas hinnoittelu, valvonta ja ylläpitotyökalut olivat syy Azureen siirtymiseen. PDF-tiedostojen luonti aiheuttaa nykyisessä järjestelmässä satunnaisia kuormituspiikkejä ja siirtämällä ne Physiotoolsin sovelluspalvelimelta saadaan parannettua sovelluspalvelimen kuormituskestävyyttä. Trial-asennusten luonnissa puolestaan kutsuvat järjestelmät voivat olla hyvin erilaisia ja rajapintaan tarvitaan muunneltavuutta, joka on mahdollista toteuttaa API managementin avulla käyttämällä samaa funktiota kuhunkin tarpeeseen sopivalla tavalla muunneltuna.

3.4 Vaihtoehtoiset pilvipalvelut

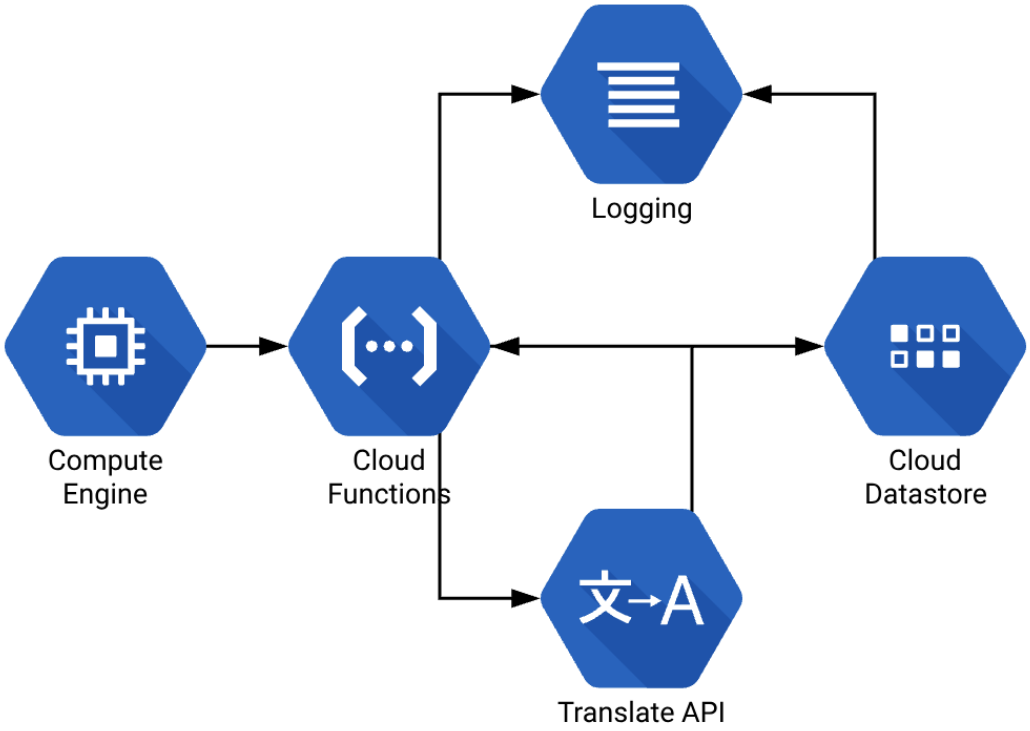
Seuraavia vaihtoehtoisia pilvipalveluita ei päädytty käyttämään, koska Azurella oli jo kyseisessä yrityksessä tehty töitä. Amazonin tarjoamaan AWS palveluun päädyttiin kuitenkin tutustumaan tarkemmin.

AWS Lambda on Amazonin tarjoama vastaava event-pohjainen palvelimeton palvelu. Yhdenmukainen palvelu Azure Durable Functionsin kanssa on Amazon Step Functions.



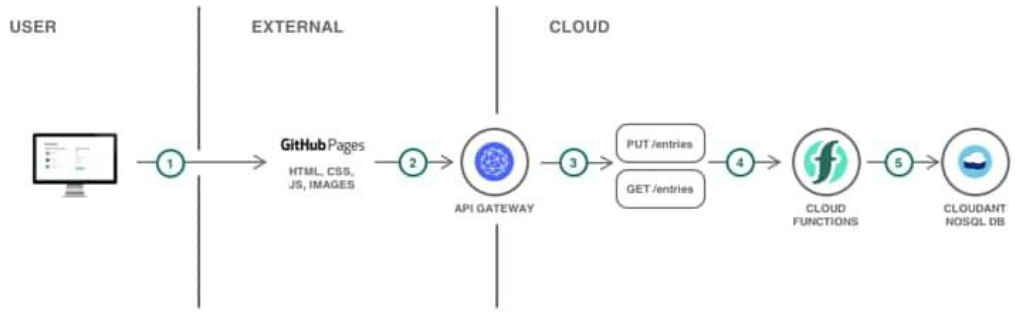
KUVIO 14. AWS Lambda Web API -palvelu. (Amazon, AWS Lambda)

Google Cloudin tarjoama Cloud Functions on toinen vastaava palvelu. Se on skaalautuva pay-as-you-go FaaS-palvelu, joka helpottaa event-pohjaisten palvelujen tekemistä.



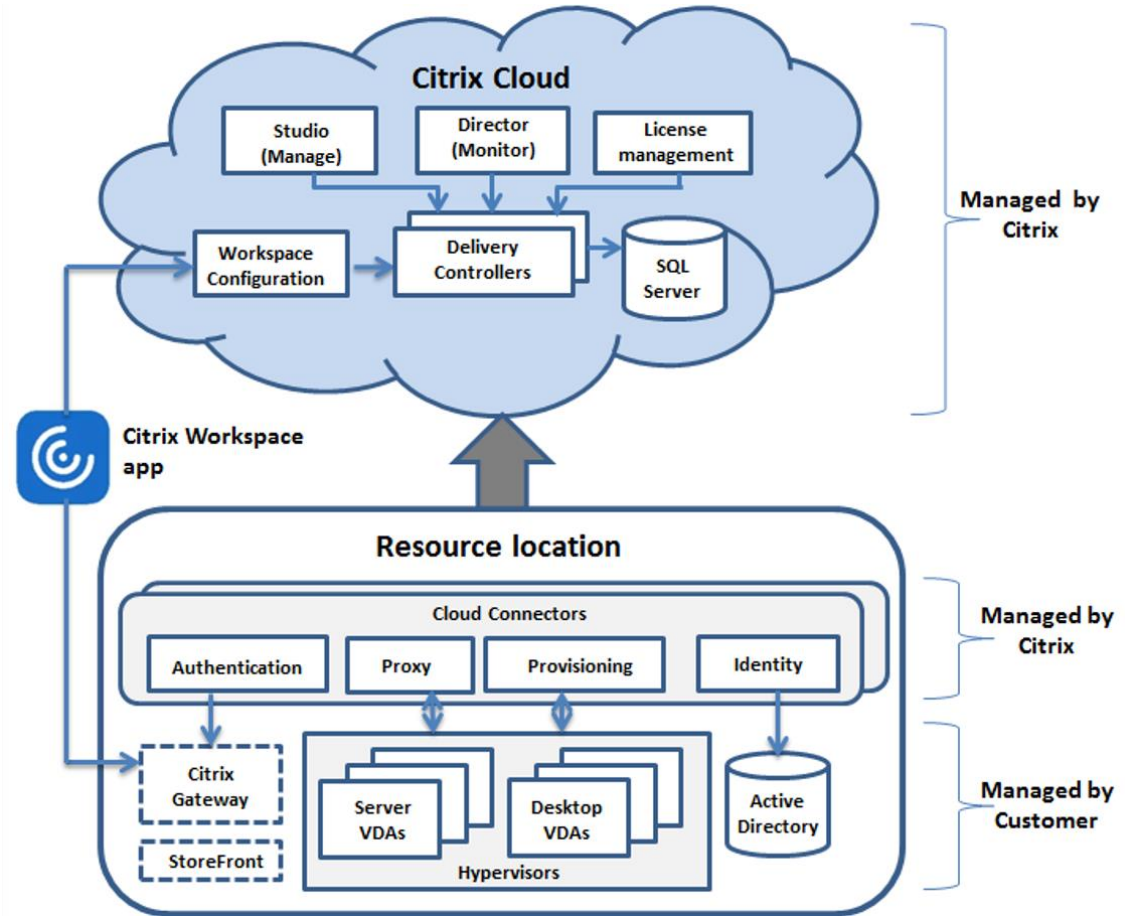
KUVIO 15. Google Cloudin tarjoama Cloud Functions -palvelu. (Liam Mackey 2019.)

IBM Cloud Functions on kolmas vastaava palvelu. Sekin on Faas-palvelu, joka tarjoaa muun muassa palvelittomat- ja mobiilibackendit, integraation muihin palveluihin ja hintatehokkaan laskennan.



KUVIO 16. IBM Cloud Functions Web API -palvelu. (IBM, Serverless web application and API.)

Citrix on neljäs vastaava pilvipalvelun tarjoaja, josta saa Cloud Connectorien avulla tehtyä vastaavia applikaatioita. Citrix huolehtii esimerkiksi SQL-serveri ja autentikointiasioista asiakkaan puolesta.



KUVIO 17. Citrix Cloudin toimintaperiaatteet (Ana Ruiz 2021.)

4 RAJAPINNAN TOTEUTUS

4.1 CreateTrial-rajapinnan on-premise -toteutustapa

REST API MVC -toteutuksessa action-metodi -kutsupyyntöjä ohjataan kontrolle-reissa, jotka pyörivät IIS -palvelimessa. Action-metodia CreateTrial kutsuttaessa se ottaa vastaan uuden installaation luomiseen tarvittavat parametrit ja palauttaa http-vastauskoodin OK tai antaa virheilmoituksen. Kaikki tämä tapahtuu installaa-tiokontrollerissa, mistä vastauskoodi palautetaan kutsujalle.

```
[HttpGet]
[Route("Api/Installation/CreateTrial")]
1 reference
public HttpResponseMessage CreateTrial(string ServiceUserId, string CompanyName, string ContactEmail, string ContactFirstName, string ContactLastName,
{
    HttpResponseMessage resultMessage;

    if (string.IsNullOrWhiteSpace(ServiceUserId) || string.IsNullOrWhiteSpace(CompanyName) || string.IsNullOrWhiteSpace(ContactEmail)
        || string.IsNullOrWhiteSpace(ContactFirstName) || string.IsNullOrWhiteSpace(ContactLastName))
    {
        resultMessage = Request.CreateResponse(HttpStatusCode.BadRequest);
        return resultMessage;
    }

    ReturnInfo result = new ReturnInfo();

    CreateParameters licenseParameters = new CreateParameters();

    licenseParameters.ServiceUserId = ServiceUserId;
    licenseParameters.CompanyName = CompanyName;
    licenseParameters.ContactEmail = ContactEmail;
    licenseParameters.ContactFirstName = ContactFirstName;
    licenseParameters.ContactLastName = ContactLastName;
    licenseParameters.Country = Country;
    licenseParameters.Language = Language;
    licenseParameters.Address = Address;
    licenseParameters.Address2 = Address2;
    licenseParameters.Address3 = Address3;
    licenseParameters.City = City;
    licenseParameters.State = State;
    licenseParameters.Zip = Zip;
    licenseParameters.Phone = Phone;
    licenseParameters.LinkSiteId = LinkSiteId;
    licenseParameters.LinkUserId = LinkUserId;

    result = Create(licenseParameters);

    resultMessage = Request.CreateResponse(result.StatusDescription);
    resultMessage.Content = new StringContent(result.ErrorMessage, Encoding.UTF8, "text/plain");

    return resultMessage;
}
```

KUVIO 18. On-premise -version installaatiokontrollerin CreateTrial-metodi

Kuten kuviosta 18 nähdään installaatiokontrollerin CreateTrial-metodissa, tarkas-tetaan ensin pakollisten parametrien sisältö. Jos parametreja puuttuu, niin palau-tetaan vastauskoodi BadRequest 400. Tämän jälkeen Create-funktioon viedään lisenssiparametrit ja muodostetaan siellä haluttu palautuskoodi ja viesti. Lopuksi sijoitetaan Create-funktiossa päätetyt arvot HttpResponseMessage muuttujaan ja annetaan kontrollerin palauttaa vastauskoodi viestin kanssa.

4.2 CreateTrial-rajapinnan Azure Function -toteussuunnitelma

Tarkoituksena on toteuttaa palvelimeton versio sovelluksesta. Kehitys ja testaus tehdään Visual Studio -ympäristössä C# -kielellä. Ulkoinen rajapinta on tarkoitus pitää samanlaisena kuin olemassa oleva järjestelmä.

Azure-portaalissa luodaan funktioille oma resurssiryhmä ja annetaan Azuren antamien ehdotusten mukaisesti sille Application Insight -palvelun tuoma monitorointi, sekä tallennustili mihin sovelluksen käyttämää dataa varastoidaan. Funktioiden koodi voidaan luoda joko Azure-portaalissa tai Visual Studio -tai Visual Studio Code -ympäristöissä.

```
[FunctionName("CreateTrial")]
0 references
public static HttpResponseMessage CreateTrial([HttpTrigger(AuthorizationLevel.Anonymous, "get", Route = "Installation/CreateTrial")]
    HttpRequest req,
    ILogger logger)
{
    string ServiceUserId = req.Query["ServiceUserId"];
    string CompanyName = req.Query["CompanyName"];
    string ContactEmail = req.Query["ContactEmail"];
    string ContactFirstName = req.Query["ContactFirstName"];
    string ContactLastName = req.Query["ContactLastName"];
    string Country = req.Query["Country"];
    string Language = req.Query["Language"];
    string Address = req.Query["Address"];
    string Address2 = req.Query["Address2"];
    string Address3 = req.Query["Address3"];
    string City = req.Query["City"];
    string State = req.Query["State"];
    string Zip = req.Query["Zip"];
    string Phone = req.Query["Phone"];
    string LinkSiteId = req.Query["LinkSiteId"];
    string LinkUserId = req.Query["LinkUserId"];

    HttpResponseMessage resultMessage;
    HttpRequestMessage requestMessage = new HttpRequestMessage();

    if (string.IsNullOrWhiteSpace(ServiceUserId) || string.IsNullOrWhiteSpace(CompanyName) || string.IsNullOrWhiteSpace(ContactEmail)
        || string.IsNullOrWhiteSpace(ContactFirstName) || string.IsNullOrWhiteSpace(ContactLastName))
    {
        resultMessage = requestMessage.CreateResponse(HttpStatusCode.BadRequest);
        return resultMessage;
    }

    ReturnInfo result = new ReturnInfo();

    CreateParameters licenseParameters = new CreateParameters();
```

KUVIO 19. Azure Functions CreateTrial-metodi

Azure Functions CreateTrial-metodin kutsu (kuvio 19) on melkein samanlainen kuin on-premise -version kutsu (kuvio 18). Kuten funktiokutsusta huomataan, toisin kuin installaatiokontrollerissa parametrit luetaan ohjelmakoodissa sen sijaan, että ne olisi kuvattu funktion parametrimäärityksissä. Funktio palauttaa installaatiokontrollerin tapaan HttpResponseMessage muuttujan paluuarvona.

4.3 Toteutusten vertailu

CreateTrial-metodoita vertailleissa huomattiin on-premise -version olevan yksinkertaisempi ohjelmoida .NET-sovelluskehyksen monipuolisten ominaisuuksien takia. Azure Functions -version toteutukseen käytetyssä .NET Core -sovelluskehyksessä ne olivat erilaisia tai puuttuivat kokonaan.

Yksittäisen API-kutsun toteuttaminen Azure Functions -versiossa puolestaan oli kevyempää ja Azuressa pystytään suoraan hyödyntämään kaikkia lisäominaisuuksia, kuten monitorointi- ja valvontatyökaluja. Azure Functions -toteutuksen laajentaminen API management -palvelun avulla tuo mahdollisuuden pienellä työllä tehdä useita versioita eri yhteistyökumppanien tarpeisiin ja hallita rajapinnan käyttöä.

Html-to-pdf On-premise -versiossa (kuvio 7) pdf-tiedostot luodaan Physiotoolsin sovelluspalvelimilla, mikä aiheuttaa satunnaisia kuormituspiikkejä palvelimille. Tämä saattaa näkyä loppukäyttäjälle käyttöliittymän hitautena. Azure Functions -toteutuksessa (kuvio 9) pdf-luonti ei vaikuta itse Physiotools sovellukseen mitenkään.

5 JOHTOPÄÄTÖKSET

Työssä saatiin selvitettyä eroavaisuuksia ja hyötyjä Azure-palveluun siirtymisessä. Etenkin MVC-applikaatioiden kuormittavat ajot ja Azuren tarjoama palvelujen monipuolisuus oli suuri syy Azureen siirtymisessä.

5.1 Eroavaisuudet

Azure on monipuolisempi alusta käyttää kuin ASP.NET MVC tai mikään muu .NET-vaihtoehto. Se tarjoaa valvonta- ja analysointityökalujen lisäksi monimuotoisia palveluja esimerkiksi SQL-tietokantojen, virtuaalikoneiden ja tietosuojaohjelmien käyttöön. SQL-tietokantoja ja muita on toki mahdollisia toteuttaa myös MVC -ympäristön kautta, mutta tarvitsevat enemmän koodausta.

5.1.1 Tehokkuus

Azure on kevytkäyttöinen ja palvelimeton vaihtoehto. Se tarjoaa monipuolisia ja monitahoisesti ajettavia palveluja funktioita käytettäessä. Yhtenä esimerkkinä tehokkuudesta on tapaus, jossa ajettiin 100 000 Azure Functions -tapahtumaa sekunnissa yhdeksän päivän ajan (Paul Batum 2017). Tänä aikana käsiteltiin yhteensä 76 biljoonaa tapahtumaa, mikä on huomattava määrä.

5.1.2 Käyttöystävällisyys

Azure kattaa hyvin suuren joukon erilaisia palveluja, jotka on koottu yhden käyttöliittymän taakse. Azure-portaali on jatkuvan kehityksen kohteena ja sieltä löytyy usein palveluita, joista on julkaistu vasta beta-versioita. Käyttöliittymän toiminnot ovat kuitenkin suhteellisen yhteneväisiä, joten uusien ominaisuuksien oppiminen on helpompaa. Vaikka kaikkea ei osaa käyttää, niin Microsoftin tarjoamista dokumenteista löytyy yksityiskohtaiset ohjeet ja esimerkit palveluiden käyttöön. Microsoft Learn tarjoaa myös yksityiskohtaisia oppimateriaaleja Azuren ja muiden

Microsoft-tuotteiden käytön opetteluun. Esimerkiksi Azure Functions -ohjelmointiin löytyy esimerkkejä, harjoitustehtäviä ja Microsoft-sertifiointiin valmistavia kursseja.

5.1.3 Ylläpito

Azure Functions ja muut Azuren palvelut saavat jatkuvasti päivityksiä ja vanhojen versioiden päivittäminen on helppoa. Päivitykset tuovat tullessaan muun muassa tehokkuutta, luotettavuutta ja tietoturvaa funktioilla tehtyihin palveluihin. Työtä tehdessä .NET 6 oli uusin pitkällä aikavälillä tukea saava SDK-paketti. Oman sovelluskoodin ylläpitoa helpottaa Githubin ja Visual Studion toimiva integrointi Azureen.

5.1.4 Kustannukset

Azure tarjoaa käyttäjälleen monia erilaisia hinnoittelumalleja ja käyttöehtoja. Ohjelmakoodin suoritukseen voi valita käyttötapauksen mukaan helposti kustannustehokkaimman ja skaalautuvimman vaihtoehdon. Halvimmillaan jotkut hinnoitteluvaihtoehdot ovat ilmaisia tai kuuluvat johonkin tiettyyn rajaan asti edullisimpiin tilauksiin. Hinnoitteluperusteena voi olla kutsujen määrä, prosessointiaika tai kiinteä hinnoittelu. Perushinnoittelun lisäksi palvelun skaalaaminen kasvattaa kustannuksia valitun skaalaustavan mukaisesti. Tiedon tallentamiseen on esimerkiksi edullinen Blob Storagen hinnoittelu, missä on tavanomaisen tallennustilin lisäksi harvakseltaan käytetyn tiedon arkistointivaihtoehto. Arkistointihinnoittelussa lukeminen tulee kalliiksi, mutta tallennustilin tietoa säilytetään edullisesti.

LÄHTEET

Microsoft. Azure Functions -esittelysivu. https://azure.microsoft.com/en-us/services/functions/

Microsoft. Azure Functions -dokumentaatiot. https://docs.microsoft.com/en-us/azure/azure-functions/

Microsoft. Microsoft Learning -sivut. https://docs.microsoft.com/en-us/learn/

Mahesh Chand. What is Azure Functions. Käyttöohje. Luettu 2.3.2022. https://www.c-sharpcorner.com/article/what-is-azure-functions/

Microsoft. Azure palveluiden hintavertailu. https://azure.microsoft.com/en-us/pricing/

Bijay Kumar. What are Azure Durable Functions. Luettu 28.3.2022. https://azurelessons.com/what-are-azure-durable-functions/

Liam Mackey. The World of Cloud Functions. Luettu 28.3.2022. https://www.iinteractive.com/notebook/2019/03/25/the-world-of-cloud-functions/

Paul Batum. Processing 100,000 events per second on Azure Functions. Luettu 5.4.2022. https://azure.microsoft.com/en-us/blog/processing-100-000-events-per-second-on-azure-functions/