AAL Programme

AAL
PROGRAMME

SAVE
SAfety of elderly people and Vicinity Ensuring

AAL-CP-2018-5-149-SAVE

01/09/2019 - 31/08/2022

AAL Programme

Project - SAfety of elderly people and Vicinity Ensuring - "SAVE"

Deliverable: D.1.1 Technology Assessment and Research Plan

Version: 2.0

Main editor: UNITBV

Contributing partners: ISS, VS

Table of contents

1. Internet of Things (IoT) general framework and criteria overview (aligning with AAL program objectives)

Institute of Space Science (ISS) - Partner 3 within the international SAVE consortium: beyond the support activities like SE assistance, Pilot Service Deployment and Validation highlighted on the well-being and e-Health sensors subassembly components, Research and Dissemination, ISS-P3 main activities within SAVE Project will aim the use of IoT in order to develop the two main components, namely: e-Health sensors subassembly component; technology club: well-being component. IoT will allow ISS-P3 to create a network of objects in order to connect different devices, services and/or automations as described in project proposal.

AAL Projects should have a Potential Impact on Market Development, thus one of the main targets of AAL Programme is to contribute by means of open interfaces and interoperability to the creation of a European market.[1] Also, AAL considers several requirements as: adaptability, open interfaces, interoperability and scalability.

Open Source Hardware (OSHD)[2] fits the requirements listed above and involves advantages as:

- readily-available components;
- standard processes;
- open infrastructure;
- unrestricted content;
- open source design.

OSHD is based on development boards equipped with modular sensors in order to interact with the environment. The best-known manufactures of OSHD existent of the market are Arduino, Raspberry PI, Microchip, National Instruments, Intel, Olimex etc. (Examples of OSHD boards can be seen in Figure 1).

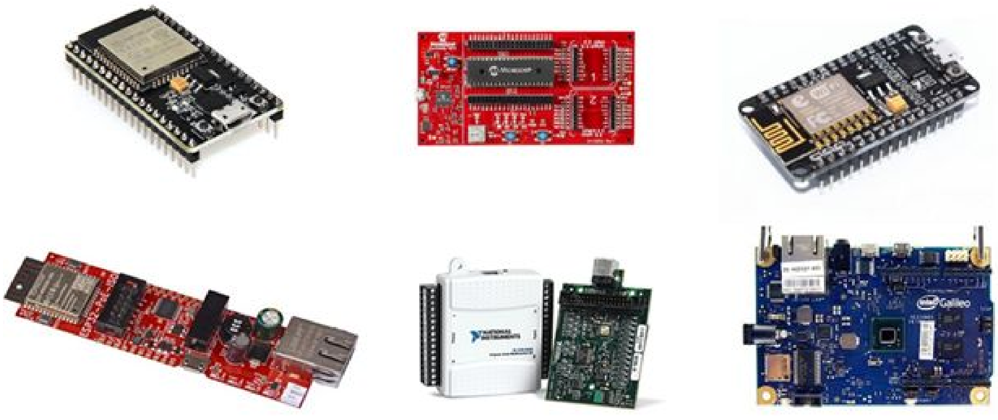


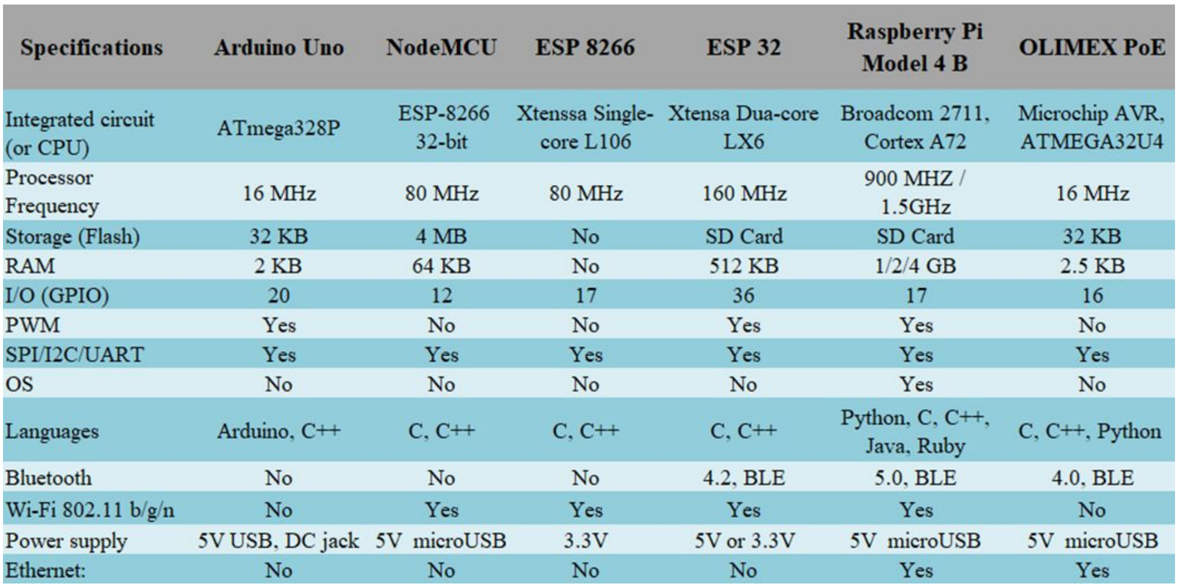
Figure 1 - Examples of OSHD boards (ESP 32, Microchip, NodMCU, OLIMEX PoE, National Instruments USB-6003, Intel Galileo)

[1] AAL guide for Applicants

[2] Open-Source Hardware Association

In order to achieve a comparative analysis of the OSHD boards existent on the market, in the Table below are presented several specifications of Arduino Uno, NodeMCU, ESP32, Raspberry Pi 4B, Olimex PoE.

Table 1 - Comparisons of OSHD boards



Specifications	Arduino Uno	NodeMCU	ESP 8266	ESP 32	Raspberry Pi Model 4 B	OLIMEX PoE
Integrated circuit (or CPU)	ATmega328P	ESP-8266 32-bit	Xtenssa Single-core L106	Xtensa Dua-core LX6	Broadcom 2711, Cortex A72	Microchip AVR, ATMEGA32U4
Processor Frequency	16 MHz	80 MHz	80 MHz	160 MHz	900 MHZ / 1.5GHz	16 MHz
Storage (Flash)	32 KB	4 MB	No	SD Card	SD Card	32 KB
RAM	2 KB	64 KB	No	512 KB	1/2/4 GB	2.5 KB
I/O (GPIO)	20	12	17	36	17	16
PWM	Yes	No	No	Yes	Yes	No
SPI/I2C/UART	Yes	Yes	Yes	Yes	Yes	Yes
OS	No	No	No	No	Yes	No
Languages	Arduino, C++	C, C++	C, C++	C, C++	Python, C, C++, Java, Ruby	C, C++, Python
Bluetooth	No	No	No	4.2, BLE	5.0, BLE	4.0, BLE
Wi-Fi 802.11 b/g/n	No	Yes	Yes	Yes	Yes	No
Power supply	5V USB, DC jack	5V microUSB	3.3V	5V or 3.3V	5V microUSB	5V microUSB
Ethernet:	No	No	No	No	Yes	Yes

ISS-P3 integration main approach in SAVE project, will be mainly oriented on OSHD technology, towards the two main directions: e-Health sensor sub-assembly component; technology club: well-being component, as shown along section 2 in the SAVE project proposal.

2. Definition of technology

Definition of a technology *(within the fields of e-Health sensors component and technology club well-being component)*	
Title	1. Stress assessment Technologies 2. Actigraphy based Technologies 3. e-Health sensors Technologies 4. Communication Technologies

Description	1. **Stress assessment Technologies** are based on wearable devices measuring blood pressure and galvanic skin response (GSR), but also temperature, and cortisol in order to evaluate specific changes and detect stress level and phase (excitement, stress and recovery). Stress assessment Technologies give a way to find a balance between work and free time in order to enhance the well-being state of the person. 2. **Actigraphy based Technologies** are based on wearable devices monitoring human rest/activity cycles. Usually, the trackers are packed as a watch that should be worn continuously in order to record data that can be analyzed offline or in real time. Also, the trackers use sensors for temperature, ambient light sound levels, skin resistance or parkinsonian tremor. 3. **e-Health sensors Technologies** provide information on disease prevention, early-detection of symptoms and physiological and psychometric monitoring at home. The elder's condition, based on physiologic and/or psychometric parameters. The e-Health devices can monitor heart rate, oxygen saturation, blood pressure, and other parameters of interest in order to keep track of elder's health and the elder's physical condition based on e-Health parameters. 4. **Communication Technologies** are based mostly on Internet of Things (IoT), the main concept in e-Health systems, which combines wearable sensors, communication systems, and mobile user interfaces. e-Health systems are based on Cloud infrastructure, offering to both elders and caregivers the possibility to track the medical history of the elder.
Target group (end users / institutions, elder people / caregivers / doctors, etc)	1. Elder people, Caregivers; 2. Elder people, Caregivers; 3. Elder people, Caregivers; 4. Institutions, Caregivers, Social and medical insurance services, Companies for elderly people care, Medical centers and general practitioner.
Scenario of use	1. The elder wears the stress assessment device that analyses the evolution of his stress level. The elder can be advised to stop working, relax and/or breathe in order to diminish stress level and thus, to enhance its well-being state 2. The elder wears the actigraphy sensor that analyses the elder's rest/activity cycles. Using the embedded sensors, the tracker notifies the subject that he is inactive in the last hour and he should

	walk or do some physical exercises. The device also tracks the number of steps and estimates the calories burned during the day. 3. The elder tests periodically its e-Health parameters. The elder will be pre-trained for proper use of each e-Health device, by a caregiver and/or a volunteer. After the tests are done, the data is stored in the cloud, and both elder and caregiver will be able to track anytime the e-Health history. 4. The devices used for elder monitoring are connected and communicating with a cloud service. The elder, or the caregiver can access the data to observe the elder history and activity.
Involved devices	1. Smart watches, Smart Bracelets, Smart Rings 2. Smart watches, Smart Bracelets, Smart Rings 3. e-Health sensors 4. IoT devices
Methods for assessment	The technologies in each category available on the market will be assessed from different points of view, considering the project requirements, user requirements, ease of use, compatibility and price

3. The context of e-Health sensors (subassembly) component

Commercial off the shelf (COTS*) devices identification and comparative analysis

Table 2 - Comparisons of COTS devices for Technology Club component

No	Company Name	Product Name	Heartrate	Blood Pressure	SpO_2	Body Position	Body Temp.	Galvanic Skin Response	Air flow	EKG	Snore	Spirometer	Accelerometer	GPS	Android App	Photo	Link
1.	cooking Hacks	e-Health v2	Y	-	Y	Y	Y	Y	Y	-	-	-	-	-	-		https://www.cooking-hacks.com/mysignals-hw-ehealth-medical-biometric-arduino-complete-kit-ble
2.	Sleepon	Go2Sleep	Y	-	Y	-	-	-	Y	-	-	-	Y	-	Y		https://www.sleepon.us/
3.	cooking Hacks	My Signals Hardware	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			-		https://www.cooking-hacks.com/mysignals-hw-ehealth-medical-biometric-arduino-complete-kit-ble
4.	cooking Hacks	My Signals Software	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			-		https://www.cooking-hacks.com/mysignals-sw-ehealth-medical-biometric-complete-kit-ble/

5.	Withings	Watch + BPM care	-	Y	-	-	-	-	-	-	-	-			-		https://www.withings.com/ro/en/bpm-core/shop
6.	Masimo	Ras97	Y	Y	Y	-	Y	-	-	Y	-	-			-		https://www.masimo.com/products/continuous/rad97-family/
7.	Comarch	e-Care Band	Y	-	-	Y	-	-	-	-	Y	-			Y		https://www.comarch.com/healthcare/products/remote-medical-care/remote-care-services/e- careband/
8.	Comarch	Cardio-Vest System	Y	-	-	-	-	-	-	-	-	-			-		https://www.comarch.com/healthcare/products/remote-medical-care/remote-care-services/e-careband/

4. The context of well-being component

Well-being interconnects the dimensions of physical, mental, and social well-being as extend beyond the traditional definition of health and being characterized by positive or pleasant emotions ranging from contentment to intense joy.

Well-being is determined by choices and activities aimed at achieving physical vitality, mental alacrity, social satisfaction, a sense of accomplishment, and personal fulfilment. Nevertheless, **well-being** could also be a condition of a group, thus a high level of well-being means that in some sense the group's condition is positive.

The Technology Club should help and answer our main topic about elderly well-being, using the benefits of physical and mental activities, improving their ability to adapt to the environment and to improve their social relations.

In our opinion, the Technology Club - Well-being component has four main aspects:

1. **Activities aspect**, the elders could perform with/without the caregivers/volunteer's moderate physical exercise, mental and social activities.
2. **Environment aspect**, the end-users could spend quality time together in a dedicated physical or virtual location;
3. **Sensors aspect**, the end-user can be monitored, at the Club, through video, environmental and wearable sensors; for example, after social activities, the user's well-being (mental aspect) can be measured with a stress sensor;
4. **Integration aspect**

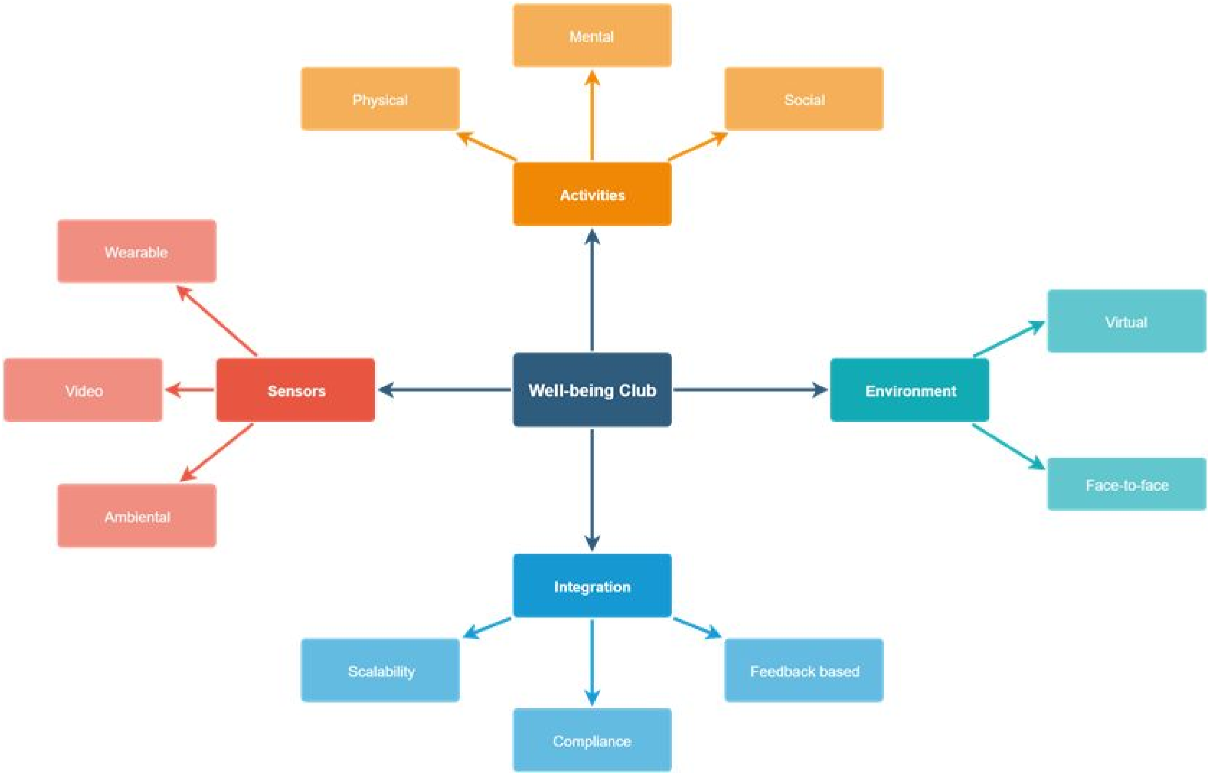


Figure 2- Aspects of Technology Club - Well-being component

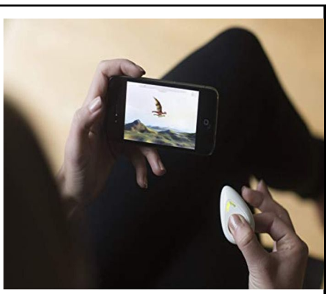
Technology Club development shall imply an integration of existing fitness devices COTS-Commercial Off The Shelf, as those presented in the table above, and implementation of relevant existing methodologies for improving/maintaining well-being of the end-users.

5. Commercial off the shelf COTS devices identification and comparative analysis

For implementing well-being component, in the Table below we presented a list of Commercial off the shelf (COTS) devices which will allow the developer to select the most suitable devices by a comparative analysis planned for the future phase.

Table 2 - Comparisons of COTS devices for well-being component

No	Product name	Product Type	Producer	Description	Link	Image
1.	ActiGraph HT9X Link	actigraphy watch	ActiGraph Corp	Includes a gyroscope, magnetometer, and secondary accelerometer to deliver valuable information about movement, rotation, and body position in sleep and day activity monitoring.	https://www.actigraphcorp.com/actigraph-link/	 https://s3.amazonaws.com/actigraphcorp.com/wp-content/uploads/2018/02/26184932/link-wrist-band-550x550.jpg
2.	WellShell	fitness device	TaoWellnes	50 isometric exercises can be performed, for different body parts, with WellShell; the unit works with an app on the user's Android or iOS Bluetooth	https://newatlas.com/tao-wellshell-isometric-exercise/30324/	 https://newatlas.com/tao-wellshell-isometric-exercise/30324/

3.	Pip-Personal Stress Sensor & Mental Fitness Trainer	Stress Sensor & Mental Trainer	The Pip	Measures Galvanic Skin Response (GSR) in order to evaluate the changes in sweat gland activity that are reflective of the intensity of our emotional state, otherwise known as emotional arousal.	https://www.amazon.com/Pip-Personal-Stress-Fitness-Trainer/dp/B00MBC9KZG	 https://www.amazon.com/Pip-Personal-Stress-Fitness-Trainer/dp/B00MBC9KZG
4.	Jitterbug Smart2	Flip phone for seniors	Greatcall	Is a smartphone for seniors with simple navigation	https://www.theseniorlist.com/cell-phones/jitterbug/reviews/	 https://www.theseniorlist.com/cell-phones/jitterbug/reviews/
5.	Moodmetric	Smart Ring	Moodmetric	A Tool designed for preventive stress management allowing the user to find the balance between work and free time.	https://www.moodmetric.com	 https://www.moodmetric.com
6.	Shimmer3 GSR+	Stress Sensor	Shimmer Sensing	suitable for measuring the electrical characteristics or conductance of the skin, as well as capturing the Optical Pulse/PPG (Photoplethysmogram) signal and converting it to estimate heart rate (HR)	https://www.shimmersensing.com/products/shimmer3-wireless-gsr-sensor	 https://www.shimmersensing.com/assets/images/content/product_images/GSRWHITE_edited-5.jpg

7.	SymGym	Fitness platform	SymGym	SymGym is a training system that integrates exercise, video games, and social network into one fitness solution.	https://www.symgym.fit	 https://www.symgym.fit
8.	Samsung Fit	Smart bracelet	Samsung	It is a fitness bracelet that can monitor a wide range of sport parameters and can measure the quality of sleep and stress accumulated during the day. The device charging is wireless.	https://www.emag.ro/bratara-fitness-samsung-galaxy-fit-black-sm-r370nzkarom/pd/D714YZBBM/?X-Search-Id=8e235578a33a1f300eb5&X-Product-Id=45530382&X-Search-Page=1&X-Search-Position=2&X-Section=search&X-MB=0&X-Search-Action=view	
9.	Xiaomi Mi Band 4	Smart bracelet	Xiaomi	The new bracelet Mi Band 4 is recommended and high sports performance. The device has all the necessary applications at a small size.	https://www.emag.ro/bratara-fitness-xiaomi-mi-band-4-6934177710377/pd/DVG5SRBBM/?ref=	

10.	Hexiwear	Smart watch	Micro Electronika	Hexiwear can measure temperature and humidity in addition to sports activity. It has a docking station that offers the possibility to connect more devices.	https://www.hexiwear.com/hardware/	
11.	Samsung Watch	Smart watch	Samsung	The smart watch from Samsung has many possibilities for personalization and becomes an unique device from person to person, which makes it superior to the other devices on the market.	https://www.samsung.com/ro/wearables/galaxy-watch-r800/	
12.	Fitbit Versa	Smart watch	Fitbit	Versa is a new device that can combine elegance with utility and performance. The device reminds you to perform breathing exercises when you are stressed.	https://www.emag.ro/ceas-smartwatch-fitbit-versa-2-nfc-black-carbon-fb507bkbk/pd/DZPT06BBM/	

6. Cloud services from Microsoft Azure

Azure is a complete cloud platform that can host your existing application infrastructure, provide compute-based services tailored for your application development needs, or even augment your on-premises applications. Azure integrates the cloud services that you need to develop, test, deploy, and manage your applications—while taking advantage of the efficiencies of cloud computing. By hosting your applications in Azure, you can start small and easily scale your application as your customer demand grows. Azure also offers the reliability that's needed for high-availability applications, even including failover between different regions. You can easily manage all of your services remotely from one location [1].

Azure provides several cloud-based compute offerings to run your application so that you don't have to worry about the infrastructure details. You can easily scale up or scale out your resources as your application usage grows. Azure offers services that cover the spectrum—from providing infrastructure-as-a-service (IaaS) to give you full control over your application hosting, through platform-as-a-service (PaaS) where the services are hosted for you, to true serverless hosting where all you need to do is write your code.

6.1. App service

App Service is a platform-as-a-service (PaaS) offering of Microsoft Azure. Create web and mobile apps for any platform or device. Integrate your apps with SaaS solutions, connect with on-premises applications, and automate your business processes. Azure runs your apps on fully managed virtual machines (VMs), with your choice of shared VM resources or dedicated VMs.

App Service includes the web and mobile capabilities that we previously delivered separately as Azure Websites and Azure Mobile Services. It also includes new capabilities for automating business processes and hosting cloud APIs. As a single integrated service, App Service lets you compose various components -- websites, mobile app back ends, RESTful APIs, and business processes -- into a single solution.

Features and capabilities of App Service:

- *Multiple languages and frameworks* - App Service has first-class support for ASP.NET, Node.js, Java, PHP, and Python. You can also run Windows PowerShell and other scripts or executables on App Service VMs.
- *DevOps optimization* - Set up continuous integration and deployment with Visual Studio Team Services, GitHub, or BitBucket. Promote updates through test and staging environments. Perform A/B testing. Manage your apps in App Service by using Azure PowerShell or the cross-platform command-line interface (CLI).
- *Global scale with high availability* - Scale up or out manually or automatically. Host your apps anywhere in Microsoft's global datacenter infrastructure, and the App Service SLA promises high availability.
- *Connections to SaaS platforms and on-premises data* - Choose from more than 50 connectors for enterprise systems (such as SAP, Siebel, and Oracle), SaaS services (such as Salesforce and Office 365), and internet services (such as Facebook and Twitter). Access on-premises data using Hybrid Connections and Azure Virtual Networks.
- *Security and compliance* - App Service is ISO, SOC, and PCI compliant.

- *Application templates* - Choose from an extensive list of application templates in the Azure Marketplace that let you use a wizard to install popular open-source software such as WordPress, Joomla, and Drupal.
- *Visual Studio integration* - Dedicated tools in Visual Studio streamline the work of creating, deploying, and debugging [2].

6.2. Workflow

Azure App Service is a cloud service that's designed to solve the practical problems that engineers face today. App Service focuses on providing superior developer productivity without compromising on the need to deliver applications at cloud scale.

App Service also provides the features and frameworks that are necessary for creating enterprise line-of-business applications while supporting developers with the most popular development languages (such as Microsoft .NET, Java, PHP, Node.js, and Python). With App Service, developers can:

- Build highly scalable web apps.
- Quickly build Mobile Apps back ends with a set of easy-to-use mobile capabilities such as data back ends, user authentication, and push notifications.
- Implement, deploy, and publish APIs with API Apps.
- Tie business applications together into workflows and transform data with Logic Apps.
- All app types rely on the scalable and flexible Web Apps platform, which enables developers to have an optimized full lifecycle experience from app design to app maintenance. The lifecycle capabilities enable the following:
- *Quick app creation*. Start from scratch or pick an operational support system (OSS) package from the Azure Marketplace.
- *Continuous deployment*. Automatically deploy new code from popular source control solutions such as TFS, GitHub, and Bitbucket, and sync content from online storage services such as OneDrive and Dropbox.
- *Test in production*. Smoothly create pre-production environments and manage the amount of traffic that's going to them. Debug in the cloud when needed, and roll back if issues are found.
- *Running asynchronous tasks and batch jobs*. Run code in a background process or activate your code based on events (such as messages landing in an Azure Storage queue) and scheduled times (CRON).
- *Scaling the app*. Use one of many options to automatically scale your service horizontally and vertically based on traffic and resource utilization. Configure private environments that are dedicated to your apps.
- *Maintaining the app*. Use many of the debugging and diagnostics features to stay ahead of problems and to efficiently resolve them either in real time (with features such as auto-healing and live debugging) or after the fact by analysing logs and memory dumps.

As a whole, App Service capabilities enable developers to focus on their code and reach a stable, highly scalable production state quickly. With the API Apps and Logic Apps features, developers can build real-world enterprise applications that bridge barriers between business solutions as well as on-premises to cloud integration [3].

6.3. App types in App Service

App Service offers several *app types*, each of which is intended to host a specific kind of workload:

- *Web Apps* - For hosting websites and web applications.
- *Mobile Apps* For hosting mobile app back ends.
- *API Apps* - For hosting RESTful APIs.
- *Logic Apps* - For automating business processes and integrating systems and data across clouds without writing code.

The word *app* here refers to the hosting resources dedicated to running a workload. Taking “web app” as an example, you’re probably accustomed to thinking of a web app as both the compute resources and application code that together deliver functionality to a browser. But in App Service a *web app* is the compute resources that Azure provides for hosting your application code. If your application is composed of a web front end and a RESTful API back end, you could deploy both to a web app or you could deploy your front-end code to a web app and your back-end code to an API app. Your application may be composed of multiple App Service apps of different kinds [4].

6.3.1. Web apps

App Service Web Apps is a fully managed compute platform that is optimized for hosting websites and web applications. This platform-as-a-service (PaaS) offering of Microsoft Azure lets you focus on your business logic while Azure takes care of the infrastructure to run and scale your apps.

Web app in App Service:

In App Service, a *web app* is the compute resources that Azure provides for hosting a website or web application.

The compute resources may be on shared or dedicated virtual machines (VMs), depending on the pricing tier that you choose. Your application code runs in a managed VM that is isolated from other customers.

Your code can be in any language or framework that is supported by Azure App Service, such as ASP.NET, Node.js, Java, PHP, or Python. You can also run scripts that use PowerShell and other scripting languages in a web app [5].

6.4. Cloud services

Cloud Services is an example of Platform-as-a-Service (PaaS). Like App Service, this technology is designed to support applications that are scalable, reliable, and cheap to operate. Just like an App Service is hosted on VMs, so too are Cloud Services, however, you have more control over the VMs. You can install your own software on Cloud Service VMs and you can remote into them.

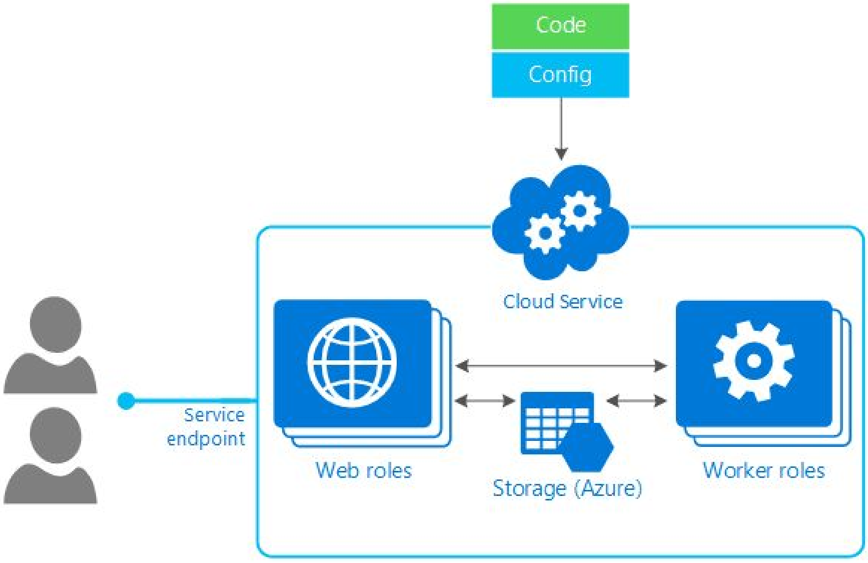


Figure 3 - Cloud services architecture

More control also means less ease of use; unless you need the additional control options, it's typically quicker and easier to get a web application up and running in Web Apps in App Service compared to Cloud Services.

The technology provides two slightly different VM options: instances of *web roles* run a variant of Windows Server with IIS, while instances of *worker roles* run the same Windows Server variant without IIS. A Cloud Services application relies on some combination of these two options.

Any combination of these two slightly different VM hosting options are available in a cloud service:

- *Web role* **-** Runs Windows Server with your web app automatically deployed to IIS.
- *Worker role* **-** Runs Windows Server without IIS.

For example, a simple application might use just a web role, while a more complex application might use a web role to handle incoming requests from users, then pass the work those requests create to a worker role for processing. (This communication could use Service Bus or Azure Queues.)

As the figure above suggests, all of the VMs in a single application run in the same cloud service. Because of this, users access the application through a single public IP address, with requests automatically load balanced across the application's VMs. The platform will scale and deploy the VMs in a Cloud Services application in a way that avoids a single point of hardware failure.

Even though applications run in virtual machines, it's important to understand that Cloud Services provides PaaS, not IaaS. Here's one way to think about it: With IaaS, such as Azure Virtual Machines, you first create and configure the environment your application will run in, then deploy your application into this environment. You're responsible for managing much of this world, doing things such as deploying new patched versions of the operating system in each VM. In PaaS, by contrast, it's as if the environment already exists. All you have to do is deploy your application. Management of the platform it runs on, including deploying new versions of the operating system, is handled for you.

6.4.1. Scaling and management

With Cloud Services, you don't create virtual machines. Instead, you provide a configuration file that tells Azure how many of each you'd like, such as *three web role instances and two worker role instances*, and the platform creates them for you. You still choose what size those backing VMs should be, but you don't explicitly create them yourself. If your application needs to handle a greater load, you can ask for more VMs, and Azure will create those instances. If the load decreases, you can shut those instances down and stop paying for them.

A Cloud Services application is typically made available to users via a two-step process. A developer first uploads the application to the platform's staging area. When the developer is ready to make the application live, they use the Azure Management Portal to request that it be put into production. This switch between staging and production can be done with no downtime, which lets a running application be upgraded to a new version without disturbing its users.

6.4.2. Monitoring

Cloud Services also provides monitoring. Like Azure Virtual Machines, it will detect a failed physical server and restart the VMs that were running on that server on a new machine. But Cloud Services also detects failed VMs and applications, not just hardware failures. Unlike Virtual Machines, it has an agent inside each web and worker role, and so it's able to start new VMs and application instances when failures occur.

The PaaS nature of Cloud Services has other implications, too. One of the most important is that applications built on this technology should be written to run correctly when any web or worker role instance fails. To achieve this, a Cloud Services application shouldn't maintain state in the file system of its own VMs. Unlike VMs created with Azure Virtual Machines, writes made to Cloud Services VMs aren't persistent; there's nothing like a Virtual Machines data disk. Instead, a Cloud Services application should explicitly write all state to SQL Database, blobs, tables, or some other external storage. Building applications this way makes them easier to scale and more resistant to failure, both important goals of Cloud Services [6].

6.5. Virtual machines

Microsoft Azure is a growing collection of integrated public cloud services including analytics, Virtual Machines, databases, mobile, networking, storage, and web — ideal for hosting your solutions. Microsoft Azure provides a scalable computing platform that allows you to only pay for what you use, when you want it - without having to invest in on-premises hardware. Azure is ready when you are to scale your solutions up and out to whatever scale you require to service the needs of your clients.

6.5.1. Azure Virtual Machines & Instances

Microsoft Azure supports running a number of popular Linux distributions provided and maintained by a number of partners. You will find distributions such as Red Hat Enterprise, CentOS, Debian, Ubuntu, CoreOS, RancherOS, FreeBSD, and more in the Azure

Marketplace. We actively work with various Linux communities to add even more flavors to the Azure endorsed Linux Distros list.

If your preferred Linux distro of choice is not currently present in the gallery, you can "Bring your own Linux" VM by creating and uploading a Linux VHD in Azure.

Azure virtual machines allow you to deploy a wide range of computing solutions in an agile way. You can deploy virtually any workload and any language on nearly any operating system - Windows, Linux, or a custom created one from any one of our growing list of partners. Still don't see what you are looking for? Don't worry - you can also bring your own images from on-premises.

6.5.2. VM Sizes

When you deploy a VM in Azure, you are going to select a VM size within one of our series of sizes that is suitable to your workload. The size also affects the processing power, memory, and storage capacity of the virtual machine. You are billed based on the amount of time the VM is running and consuming its allocated resources. A complete list of sizes of Virtual Machines.

Here are some basic guidelines for selecting a VM size from one of our series (A, D, DS, G and GS).

- A-series VMs are our value priced entry-level VMs for light workloads and Dev/Test scenarios. They are widely available in all regions and can connect and use all standard resources available to virtual machines.
- A-series sizes (A8 - A11) are special compute intensive configurations suitable for high-performance computing cluster applications.
- D-series VMs are designed to run applications that demand higher compute power and temporary disk performance. D-series VMs provide faster processors, a higher memory-to-core ratio, and a solid-state drive (SSD) for the temporary disk.
- Dv2-series, is the latest version of our D-series, features a more powerful CPU. The Dv2-series CPU is about 35% faster than the D-series CPU. It is based on the latest generation 2.4 GHz Intel Xeon® E5-2673 v3 (Haskell) processor, and with the Intel Turbo Boost Technology 2.0, can go up to 3.2 GHz. The Dv2-series has the same memory and disk configurations as the D-series.
- G-series VMs offer the most memory and run on hosts that have Intel Xeon E5 V3 family processors.

6.5.3. Automation

To achieve a proper DevOps culture, all infrastructure must be code. When all the infrastructure lives in code it can easily be recreated (Phoenix Servers). Azure works with all the major automation tooling like Ansible, Chef, SaltStack, and Puppet. Azure also has its own tooling for automation:

- Azure Templates
- Azure VMAccess

Azure is rolling out support for cloud-init across most Linux Distros that support it. Currently Canonical's Ubuntu VMs are deployed with cloud-init enabled by default. RedHats RHEL, CentOS, and Fedora support cloud-init, however the Azure images maintained by RedHat do not have cloud-init installed. To use cloud-init on a RedHat family OS, you must create a custom image with cloud-init installed.

6.5.4. Partners

Microsoft works closely with our partners to ensure the images available are updated and optimized for an Azure runtime. For more information on our partners check their marketplace pages below.

- Linux on Azure - Endorsed Distributions
- Redhat - Azure Marketplace - RedHat Enterprise Linux 7.2
- Canonical - Azure Marketplace - Ubuntu Server 16.04 LTS
- Debian - Azure Marketplace - Debian 8 "Jessie"
- FreeBSD - Azure Marketplace - FreeBSD 10.3
- CoreOS - Azure Marketplace - CoreOS (Stable)
- RancherOS - Azure Marketplace - RancherOS
- Bitnami - Bitnami Library for Azure
- Mesosphere - Azure Marketplace - Mesosphere DC/OS on Azure
- Docker - Azure Marketplace - Azure Container Service with Docker Swarm
- Jenkins - Azure Marketplace - CloudBees Jenkins Platform [7]

6.6. Azure and Internet of Things

Azure IoT Hub is a key building block when you implement this IoT solution architecture using Azure services. IoT Suite provides complete, end-to-end, implementations of this architecture for specific IoT scenarios. For example:

- The *remote monitoring* solution enables you to monitor the status of devices such as vending machines.
- The *predictive maintenance* solution helps you to anticipate maintenance needs of devices such as pumps in remote pumping stations and to avoid unscheduled downtime.

6.6.1. IoT solution architecture

The following diagram shows a typical IoT solution architecture. The diagram does not include the names of any specific Azure services but describes the key elements in a generic IoT solution architecture. In this architecture, IoT devices collect data that they send to a cloud gateway. The cloud gateway makes the data available for processing by other back-end services from where data is delivered to other line-of-business applications or to human operators through a dashboard or other presentation device.

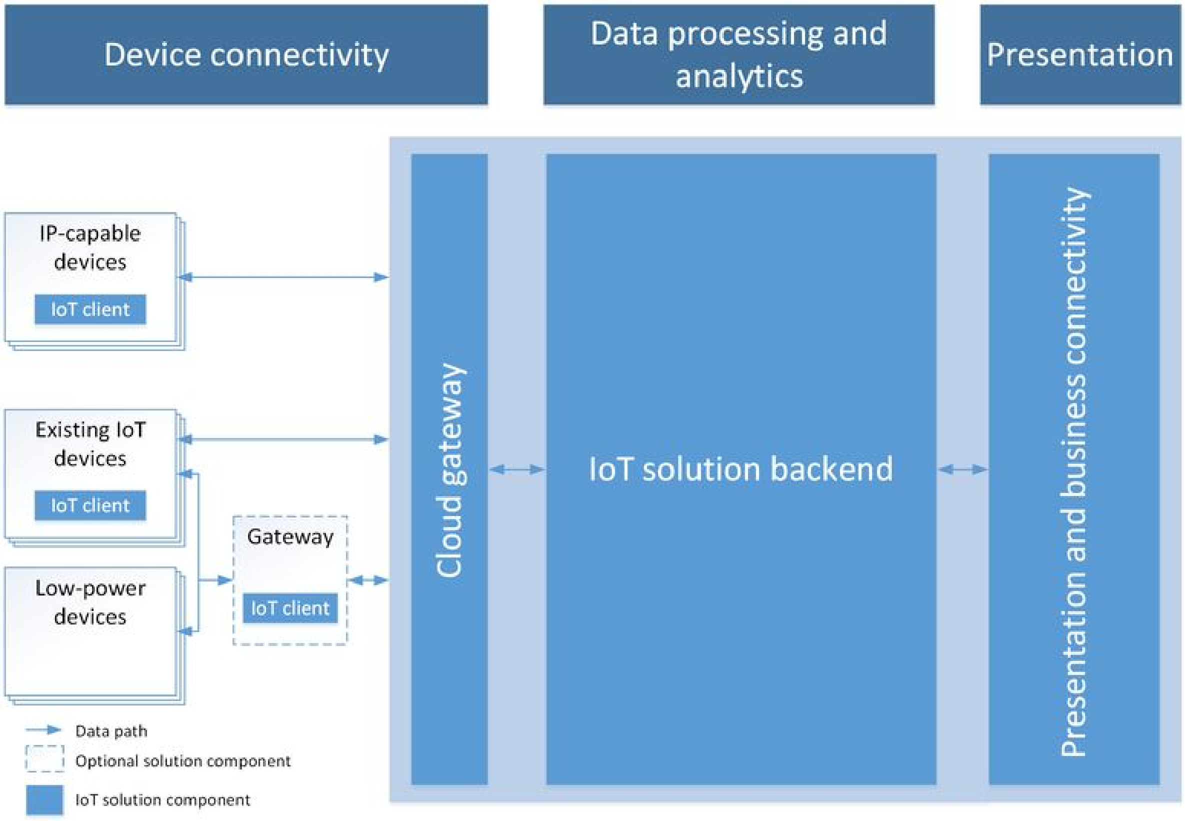


Figure 4 - Solution architecture

6.6.2. Device connectivity

In this IoT solution architecture, devices send telemetry, such as sensor readings from a pumping station, to a cloud endpoint for storage and processing. In a predictive maintenance scenario, the back end might use the stream of sensor data to determine when a specific pump requires maintenance. Devices can also receive and respond to cloud-to-device commands by reading messages from a cloud endpoint. For example, in the predictive maintenance scenario the solution back end might send commands to other pumps in the pumping station to begin rerouting flows just before maintenance is due to start to make sure the maintenance engineer can get started when she arrives.

One of the biggest challenges facing IoT projects is how to reliably and securely connect devices to the solution back end. IoT devices have different characteristics as compared to other clients such as browsers and mobile apps. IoT devices:

- Are often embedded systems with no human operator.
- Can be deployed in remote locations, where physical access is expensive.
- May only be reachable through the solution back end. There is no other way to interact with the device.
- May have limited power and processing resources.
- May have intermittent, slow, or expensive network connectivity.
- May need to use proprietary, custom, or industry-specific application protocols.
- Can be created using a large set of popular hardware and software platforms.

In addition to the requirements above, any IoT solution must also deliver scale, security, and reliability. The resulting set of connectivity requirements is hard and time-consuming to implement using traditional technologies such as web containers and messaging brokers. Azure IoT Hub and the IoT Device SDKs make it easier to implement solutions that meet these requirements.

A device can communicate directly with a cloud gateway endpoint, or if the device cannot use any of the communications protocols that the cloud gateway supports, it can connect through an intermediate gateway. For example, the Azure IoT protocol gateway can perform protocol translation if devices cannot use any of the protocols that IoT Hub supports.

6.6.3. Data processing and analytics

In the cloud, an IoT solution back end is where most of the data processing occurs, such as filtering and aggregating telemetry and routing it to other services. The IoT solution back end:

- Receives telemetry at scale from your devices and determines how to process and store that data.
- May enable you to send commands from the cloud to specific device.
- Provides device registration capabilities that enable you to provision devices and to control which devices are permitted to connect to your infrastructure.
- Enables you to track the state of your devices and monitor their activities.

In the predictive maintenance scenario, the solution back end stores historical telemetry data. The back end can use this data to use to identify patterns that indicate maintenance is due on a specific pump.

IoT solutions can include automatic feedback loops. For example, an analytics module in the back end can identify from telemetry that the temperature of a specific device is above normal operating levels. The solution can then send a command to the device, instructing it to take corrective action.

6.6.4. Presentation and business connectivity

The presentation and business connectivity layer allows end users to interact with the IoT solution and the devices. It enables users to view and analyze the data collected from their devices. These views can take the form of dashboards or BI reports that can display both historical data or near real-time data. For example, an operator can check on the status of particular pumping station and see any alerts raised by the system. This layer also allows integration of the IoT solution back end with existing line-of-business applications to tie into enterprise business processes or workflows. For example, the predictive maintenance solution can integrate with a scheduling system that books an engineer to visit a pumping station when the solution identifies a pump in need of maintenance [8].

6.7. Azure IoT Hub

Azure IoT Hub is a fully managed service that enables reliable and secure bidirectional communications between millions of IoT devices and a solution back end. Azure IoT Hub:

- Provides reliable device-to-cloud and cloud-to-device messaging at scale.

- Enables secure communications using per-device security credentials and access control.
- Provides extensive monitoring for device connectivity and device identity management events.
- Includes device libraries for the most popular languages and platforms.

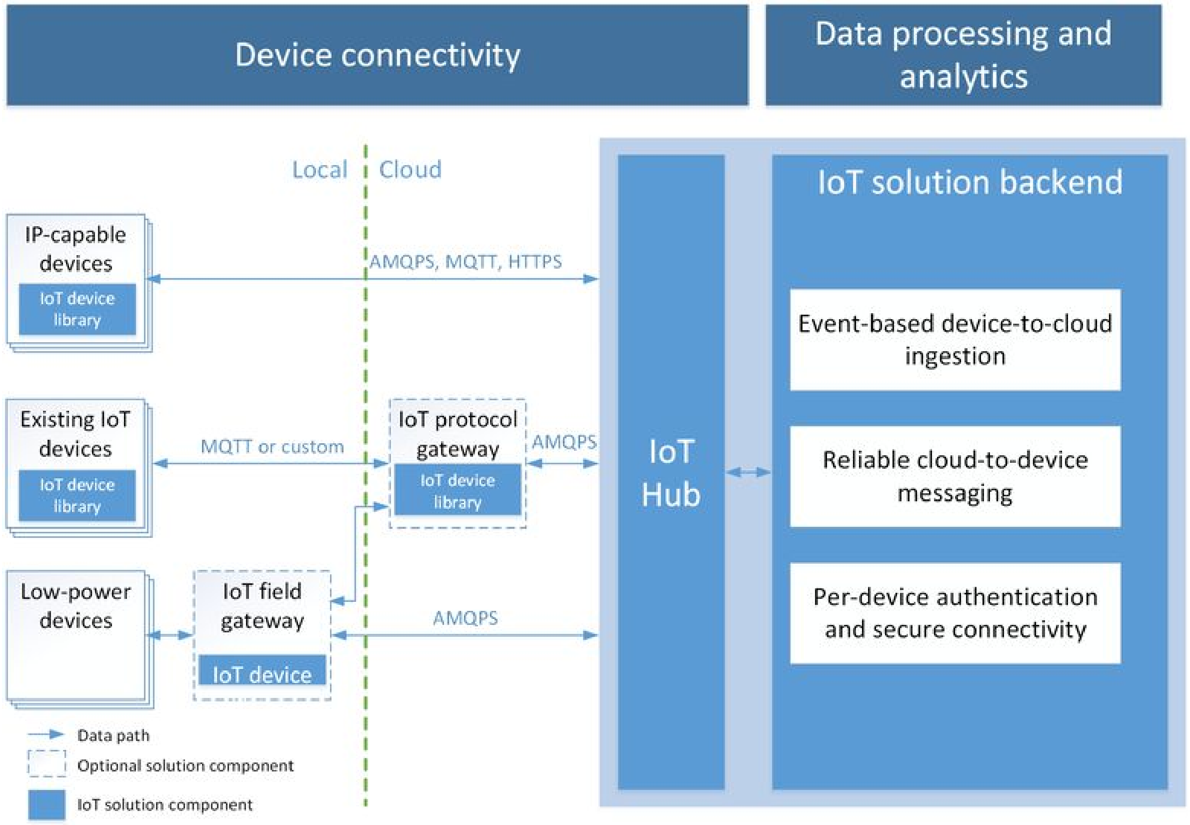


Figure 5 - Azure IoT Hub

Azure IoT Hub implements the service-assisted communication pattern to mediate the interactions between your devices and your solution back end. The goal of service-assisted communication is to establish trustworthy, bidirectional communication paths between a control system, such as IoT Hub, and special-purpose devices that are deployed in untrusted physical space. The pattern establishes the following principles:

- Security takes precedence over all other capabilities.
- Devices do not accept unsolicited network information. A device establishes all connections and routes in an outbound-only fashion. For a device to receive a command from the back end, the device must regularly initiate a connection to check for any pending commands to process.
- Devices should only connect to or establish routes to well-known services they are peered with, such as IoT Hub.
- The communication path between device and service or between device and gateway is secured at the application protocol layer.
- System-level authorization and authentication are based on per-device identities. They make access credentials and permissions nearly instantly revocable.

- Bidirectional communication for devices that connect sporadically due to power or connectivity concerns is facilitated by holding commands and device notifications until a device connects to receive them. IoT Hub maintains device-specific queues for the commands it sends.
- Application payload data is secured separately for protected transit through gateways to a particular service.
- The mobile industry has used the service-assisted communication pattern at enormous scale to implement push notification services such as Windows Push Notification Services, Google Cloud Messaging, and Apple Push Notification Service.
- IoT Hub is supported over ExpressRoute's public peering path.

6.7.1. Advantages of Azure IoT Hub

Azure IoT Hub addresses the device-connectivity challenges in the following ways:

- *Per-device authentication and secure connectivity*. You can provision each device with its own security key to enable it to connect to IoT Hub. The IoT Hub identity registry stores device identities and keys in a solution. A solution back end can add individual devices to allow or deny lists to enable complete control over device access.
- *Monitoring of device connectivity operations*. You can receive detailed operation logs about device identity management operations and device connectivity events. This monitoring capability enables your IoT solution to identify connectivity issues, such as devices that try to connect with wrong credentials, send messages too frequently, or reject all cloud-to-device messages.
- *An extensive set of device libraries*. Azure IoT device SDKs are available and supported for various languages and platforms--C for many Linux distributions, Windows, and real-time operating systems. Azure IoT device SDKs also support managed languages, such as C#, Java, and JavaScript.
- *IoT protocols and extensibility*. If your solution cannot use the device libraries, IoT Hub exposes a public protocol that enables devices to natively use the MQTT v3.1.1, HTTP 1.1, or AMQP 1.0 protocols. You can also extend IoT Hub to provide support for custom protocols by: Creating a field gateway with the Azure IoT Gateway SDK that converts your custom protocol to one of the three protocols understood by IoT Hub and customizing the Azure IoT protocol gateway, an open source component that runs in the cloud.
- *Scale*. Azure IoT Hub scales to millions of simultaneously connected devices and millions of events per second.
- These benefits are generic to many communication patterns. IoT Hub currently enables you to implement the following specific communication patterns:
- *Event-based device-to-cloud ingestion***.** IoT Hub can reliably receive millions of events per second from your devices. It can then process them on your hot path by using an event processor engine. It can also store them on your cold path for analysis. IoT Hub retains the event data for up to seven days to guarantee reliable processing and to absorb peaks in the load.
- *Reliable cloud-to-device messaging (or commands).* The solution back end can use IoT Hub to send messages with an at-least-once delivery guarantee to individual devices. Each message has an individual time-to-live setting, and the back end can

request both delivery and expiration receipts. These receipts ensure full visibility into the life cycle of a cloud-to-device message. You can then implement business logic that includes operations that run on devices.

- *Upload files and cached sensor data to the cloud***.** Your devices can upload files to Azure Storage using SAS URIs managed for you by IoT Hub. IoT Hub can generate notifications when files arrive in the cloud to enable the back end to process them [9].

7. IBM Cloud

IBM Cloud (formely know as Bluemix) is IBM's innovative cloud computing platform that combines platform as a service (PaaS) with infrastructure as a service (IaaS). Additionally, IBM Cloud has a rich catalog of cloud services that can be easily integrated with PaaS and IaaS to build business applications rapidly.

IBM Cloud has cloud deployments that fit your needs whether you are a small business that plans to scale, or a large enterprise that requires additional isolation. You can develop in a cloud without borders, where you can connect your private services to the public IBM Cloud services available from IBM. You and your team can access the apps, services, and infrastructure in IBM Cloud and use existing data, systems, processes, PaaS tools, and IaaS tools. Developers can tap into the rapidly growing ecosystem of available services and runtime frameworks to build applications using polyglot programming approaches.

With IBM Cloud, you no longer have to make large investments in hardware to test out or run a new app. Instead, we manage it all for you and only charge for what you use. IBM Cloud provides public, dedicated, and local integrated deployment models.

You can take an idea from inception, to development sandbox, to a globally distributed production environment with compute and storage infrastructure, open source platform services and containers, and software services and tools from IBM, Watson, and more. Beyond the capabilities of the platform itself, IBM Cloud also provides flexible deployment. Provision IBM Cloud resources on-premises, in dedicated private cloud environments, or in the public cloud, and manage the resources from all three types of environments in a single dashboard.

All IBM cloud resources that are deployed in public and dedicated environments are hosted from your choice of IBM Cloud Data Center locations around the world. IBM Cloud Data Centers provide regional redundancy, a global network backbone connecting all data centers and points of presence, and stringent security controls and reporting. Through IBM Cloud Data Centers, IBM can meet your most demanding expansion, security, compliance, and data residency needs.

IBM enables you to:

- Deploy high performance compute and storage infrastructure in secure IBM Cloud Data Centers around the world.
- Test and adopt a broad range of cloud services and capabilities from IBM, open-source communities, and third-party developers.
- Connect to all of your legacy systems and apps from a single, scalable, cloud platform through private network and API capabilities.

- Spin up and turn down resources in real time as your business needs or workload demands change.

7.1. Apps

The Apps dashboard provides everything you need to get your apps up and running, and to manage those apps while they run. IBM Cloud provides various boilerplates and runtimes:

A boilerplate is a template for an application and its associated runtime environment and predefined services for a specific domain.

A runtime is the set of resources that is used to run an app, provided as containers for different types of apps.

IBM Cloud provides various ways for you to run your apps, for example, Cloud Foundry and IBM Containers. Use IBM Containers to run Docker containers in a hosted cloud environment on IBM Cloud.

You can use IBM Cloud OpenWhisk for distributed, event-driven computing. OpenWhisk runs application logic in response to events or direct invocations from web or mobile apps over HTTP.

You can use IBM Cloud Mobile services to incorporate pre-built, managed, and scalable cloud services into your mobile apps.

7.2. Services

The Services dashboard provides access to the IBM Cloud services available from IBM and third-party providers. These include Watson, Internet of Things, Analytics, Mobile, and DevOps services:

Deliver innovative new applications faster and cheaper with just the right features using IBM DevOps services and the IBM Cloud Garage Method. When you adopt DevOps practices and create a culture of innovation and agility, you can iterate and pivot in response to the market.

Blockchain is a peer-to-peer distributed ledger technology for a new generation of transactional applications that establishes trust, accountability, and transparency while streamlining business processes.

Watson gives your apps the power of cognitive computing with a full suite of speech, vision, and data APIs. Solve your most complex business problems by deploying a cognitive platform with Watson services.

IBM enables you to do more with rich, integrated cloud databases and Data & Analytics services.

The IBM Internet of Things service lets your apps communicate with, and consume data that is collected by, your connected devices, sensors, and gateways. Our recipes make it easy to get devices connected to our Internet of Things cloud. Your apps can then use our real-time and REST APIs to communicate with your devices and consume the data you've set them up to collect.

IBM offers a mobile backend infrastructure where you can build multiplatform, native, or hybrid apps while also being able to monitor and test them. You can also enhance your app with analytics, security, user insight, and continuous delivery.

7.3. Infrastructure

The Infrastructure dashboard provides various services to fit your cloud infrastructure needs.

IBM Cloud infrastructure provides the highest performing cloud infrastructure available. IBM Cloud infrastructure is one platform, which takes data centers around the world that are full of the widest range of cloud computing options, then integrates and automates everything. IBM Cloud Data Centers are filled with first class computing, storage, and networking gear. Each location is built, outfitted, and operated in the same way, so you get exactly the same capabilities and availability anywhere in our footprint. Locations are connected by the industry’s most advanced network-in-a-network, which integrates distinct public, private, and internal management networks to deliver lower total networking costs, better access, and higher speed. Also, the data centers and network share a single proprietary management system. One management tool lets you control everything--every bare metal server, virtual server, and storage device--all accessible by API, portal, and mobile applications.

IBM Cloud infrastructure offers powerful bare metal servers and flexible virtual servers in a single seamless platform. All are provided on demand and billed on monthly or hourly terms. Bare metal servers provide the raw horsepower for your processor-intensive and disk I/O-intensive workloads and can be configured to your exact specifications. Virtual servers allow for high speed of deployment, flexible scalability, and pay-as-you-go billing. For high performance computing, give your cloud a boost with graphics processing unit (GPU) servers, available by the hour or monthly.

IBM Cloud infrastructure offerings are connected to a three-tiered network, segmenting public, private, and management traffic. Infrastructure on a customer's IBM Cloud account might transfer data between one another across the private network at no cost. Infrastructure offerings, such as bare metal servers, virtual servers, and cloud storage, connect to other applications and services in the IBM Cloud catalog, such as Watson services, containers, or runtimes, across the public network. Data transfer between those two types of offerings is metered and charged at standard public network bandwidth rates.

7.4. IBM Cloud - Cloud Foundry architecture

In general, you don't have to worry about the operating system and infrastructure layers when running apps on IBM Cloud in Cloud Foundry. Layers such as root filesystems and middleware components are abstracted so that you can focus on your application code. However, you can learn more about these layers if you need specifics on where your app is running.

As a developer, you can interact with the IBM Cloud infrastructure by using a browser-based user interface. You can also use a Cloud Foundry command line interface, called cf, to deploy web apps.

Clients - which can be mobile apps, apps that run externally, apps that are built on IBM Cloud, or developers that are using browsers--interact with the IBM Cloud -hosted apps.

Clients use REST or HTTP APIs to route requests through IBM Cloud to one of the app instances or the composite services.

The following figure shows the high-level IBM Cloud - Cloud Foundry architecture.

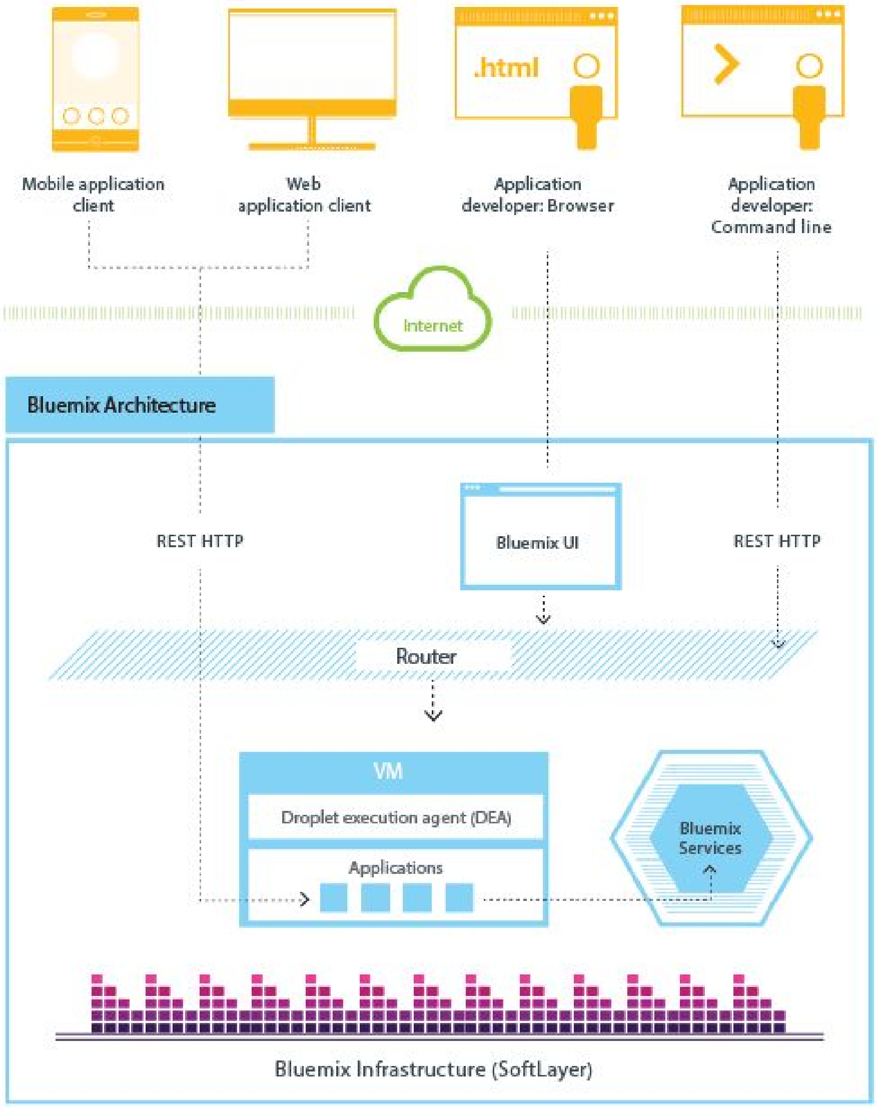

Mobile application client
Web application client
.html
Application developer: Browser
Application developer: Command line
Internet
Bluemix Architecture
REST HTTP
Bluemix UI
REST HTTP
Router
VM
Droplet execution agent (DEA)
Applications
Bluemix Services
Bluemix Infrastructure (SoftLayer)

Figure 6 - IBM Cloud Foundry architecture

You can deploy your apps to different IBM Cloud regions, for latency or security considerations. You can choose to deploy either to one region or across multiple regions.

7.5. IBM Cloud - Cloud Foundry workflow

When you deploy an app to IBM Cloud - Cloud Foundry, you must configure IBM Cloud with enough information to support the app.

For a mobile app, IBM Cloud contains an artifact that represents the mobile app's back end, such as the services that the mobile app uses to communicate with a server.

For a web app, you must ensure that information about the runtime and framework is communicated to IBM Cloud, so that IBM Cloud can set up the appropriate execution environment to run the app.

Each execution environment, including both mobile and web, is isolated from the execution environment of other apps. The execution environments are isolated even though these apps are on the same physical machine. The following figure shows the basic flow of how IBM Cloud - Cloud Foundry manages the deployment of apps:

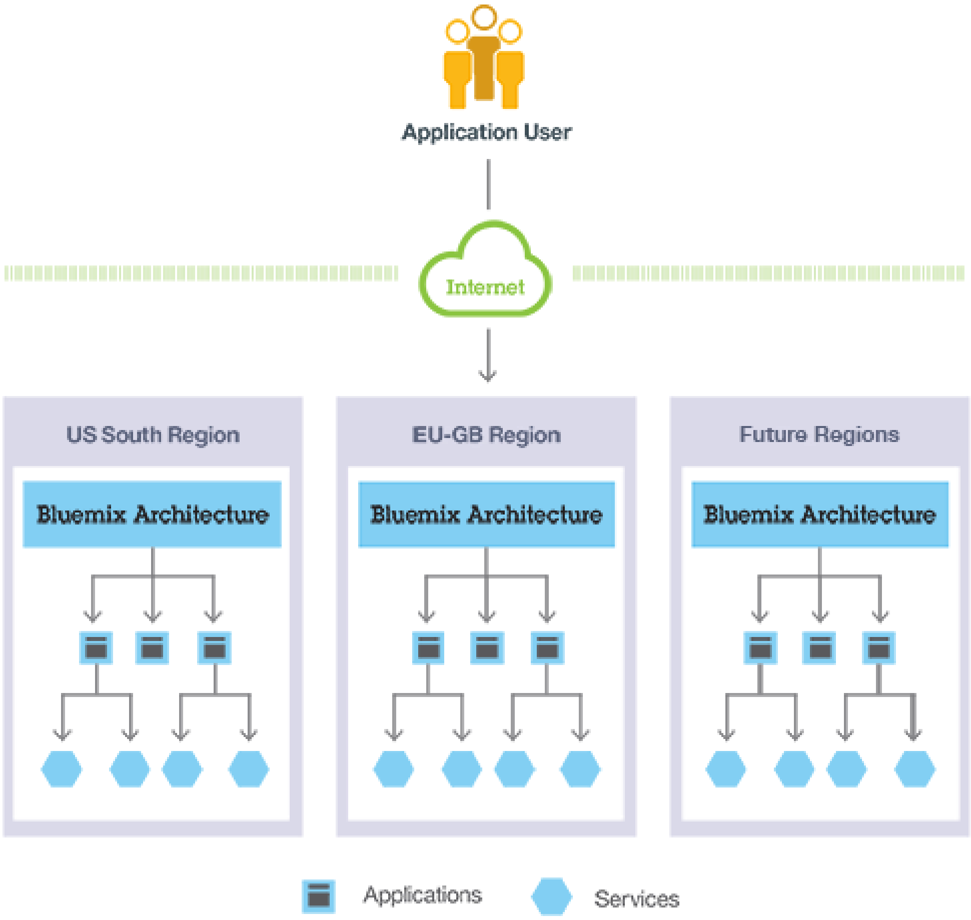


Figure 7 - Multi-region application deployment

7.6. *IBM Cloud - Cloud Foundry workflow*

When you deploy an app to IBM Cloud - Cloud Foundry, you must configure IBM Cloud with enough information to support the app.

For a mobile app, IBM Cloud contains an artifact that represents the mobile app's back end, such as the services that the mobile app uses to communicate with a server.

For a web app, you must ensure that information about the runtime and framework is communicated to IBM Cloud, so that IBM Cloud can set up the appropriate execution environment to run the app.

Each execution environment, including both mobile and web, is isolated from the execution environment of other apps. The execution environments are isolated even though these apps are on the same physical machine. The following figure shows the basic flow of how IBM Cloud - Cloud Foundry manages the deployment of apps:

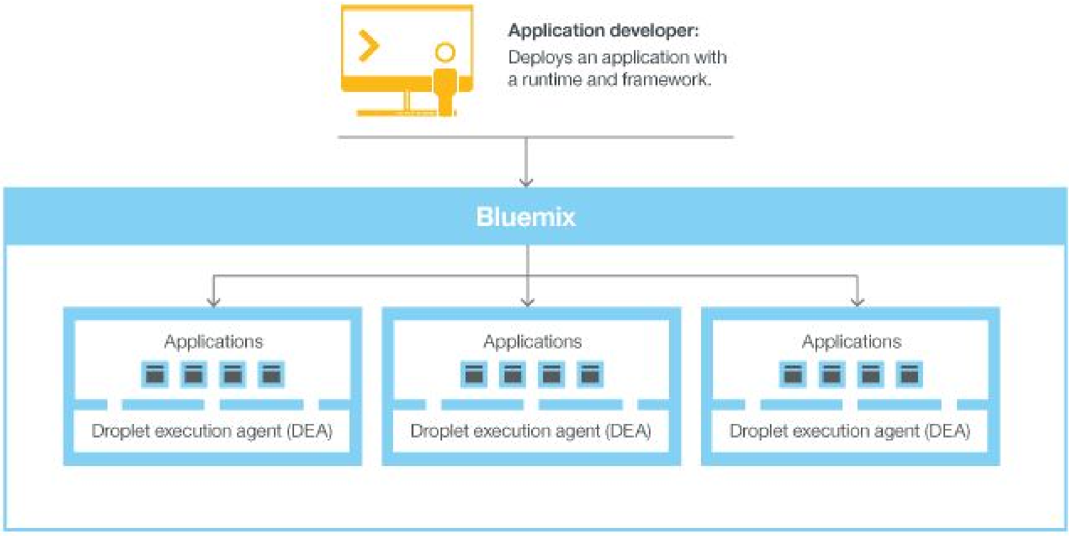

Application developer:
Deploys an application with a runtime and framework.
Bluemix
Applications
Droplet execution agent (DEA)
Applications
Droplet execution agent (DEA)
Applications
Droplet execution agent (DEA)

Figure 8 - Deploying an app

When you create an app and deploy it to IBM Cloud - Cloud Foundry, the IBM Cloud environment determines an appropriate virtual server to send the app, or the artifacts that the app represents. For a mobile app, a mobile back-end projection is created on IBM Cloud. Any code for the mobile app running in the cloud eventually runs in the IBM Cloud environment. For a web app, the code running in the cloud is the app itself that the developer deploys to IBM Cloud. The determination of the virtual server is based on several factors, including:

- The load already on the machine
- Runtimes or frameworks supported by that virtual server.

After a virtual server is chosen, an application manager on each virtual server installs the appropriate framework and runtime for the app. Then, the app can be deployed into that framework. When the deployment completes, the application artifacts are started.

The following figure shows the structure of a virtual server, also known as Droplet execution agent (DEA), that has multiple apps deployed to it.

In each virtual server, an application manager communicates with the rest of the IBM Cloud infrastructure, and manages the apps that are deployed to this virtual server. Each virtual server has containers to separate and protect apps. In each container, IBM Cloud installs the appropriate framework and runtime that are required for each app.

When the app is deployed, if it has a web interface (as for a Java web app), or other REST-based services (such as mobile services exposed publicly to the mobile app), users of the app can communicate with it by using normal HTTP requests.

Each app can have one or more URLs associated with it, but they must all point to the IBM Cloud endpoint. When a request comes in, IBM Cloud examines the request, determines which app it is intended for, then selects an instance of the app to receive the request.

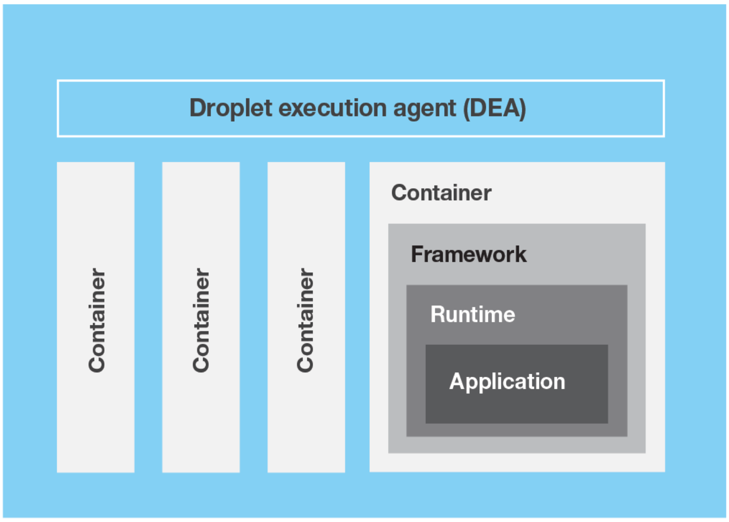


Figure 9 - Design of a virtual server

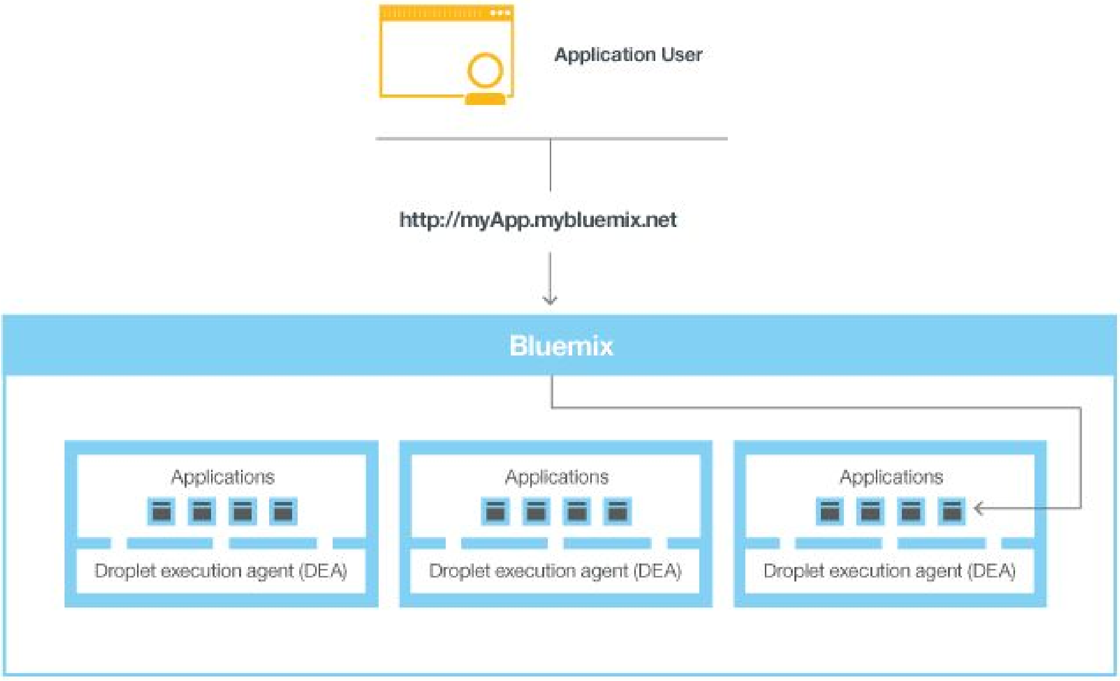

Application User
http://myApp.mybluemix.net
Bluemix
Applications
Droplet execution agent (DEA)
Applications
Droplet execution agent (DEA)
Applications
Droplet execution agent (DEA)

Figure 10 - Invoking a IBM Cloud app

7.7. Cloud Integration API

A Cloud Integration API provides secured access to the systems of record that reside behind a firewall through web APIs. When you create the Cloud Integration API, you choose the resource that you want to access through the web API, specify the operations that are permitted, and include SDKs and samples to access the API. For more information about how to create a Cloud Integration API, see Creating Cloud Integration APIs [10].

7.8. Characteristics

7.8.1. Hybrid

With public, dedicated, and local integrated deployment models, IBM Cloud represents a new way to experience cloud. IBM Cloud has cloud deployments that fit your needs, whether you are a small business that plans to scale, or a large enterprise that requires additional isolation. You can develop in a cloud without borders, where you can connect your private services to the IBM Cloud Public services that are available from IBM. And, you and your team can use existing data, systems, and processes along with the SaaS tools available in IBM Cloud.

7.8.2. Dedicated

IBM Cloud Dedicated is a private cloud with physically isolated hardware in a IBM Cloud Infrastructure (SoftLayer) data center. Single-tenant, and provisioned on a combination of bare metal and virtual servers, your IBM Cloud environment is created to fit your needs. And, with the syndicated catalog, you can power your apps with a combination of dedicated compute resources and services as well as capabilities made available through catalog syndication. Learn more about IBM Cloud Dedicated.

7.8.3. Local

With IBM Cloud Local, you can take advantage of the true value of a private cloud within your data center. IBM Cloud Local is delivered as-a-service and in constant collaboration with your IT team. It also enables you to leverage your existing mainframes, service-oriented architecture, processes, and data in a modern cloud platform. [11].

REFERENCES

[1] G. Gailey, "Microsoft," [Online]. Available: https://opbuildstorageprod.blob.core.windows.net/output-pdf-files/en-us/guides/azure-developer-guide.pdf.

[2] "Microsoft," [Online]. Available: https://docs.microsoft.com/en-us/azure/app-service/app-service-value-prop-what-is.

[3] "Microsoft," [Online]. Available: https://docs.microsoft.com/en-us/azure/app-service/app-service-how-works-readme.

[4] "Microsoft," [Online]. Available: https://docs.microsoft.com/en-us/azure/app-service/app-service-value-prop-what-is.

[5] "Microsoft," [Online]. Available: https://docs.microsoft.com/en-us/azure/app-service-web/app-service-web-overview.

[6] "Microsoft," [Online]. Available: https://docs.microsoft.com/en-us/azure/cloud-services/cloud-services-choose-me.

[7] "Microsoft," [Online]. Available: https://docs.microsoft.com/en-us/azure/virtual-machines/virtual-machines-linux-azure-overview?toc=%2fazure%2fvirtual-machines%2flinux%2ftoc.json.

[8] "Microsoft," [Online]. Available: https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-what-is-azure-iot.

[9] "Microsoft," [Online]. Available: https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-what-is-iot-hub.

[10] "IBM," [Online]. Available: https://console.ng.bluemix.net/docs/overview/whatisbluemix.html#bluemixoverview.

[1 "IBM," [Online]. Available:
1] https://console.ng.bluemix.net/docs/manageapps/depapps.html#deployingapps.