**Making Domino Applications Cloud Ready**

**BITS ZG628T: Dissertation**

by

Rajprakash Yadav

2013HT13437

**Dissertation work carried out at**

**PwC India, Kolkata**



**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE**

**PILANI (RAJASTHAN)**

November 2015

**Making Domino Applications Cloud Ready**

**BITS ZG628T: Dissertation**

by

Rajprakash Yadav

2013HT13437

**Dissertation work carried out at**

**PwC India, Kolkata**

Submitted in partial fulfillment of M.Tech. Software Systems degree programme

Under the Supervision of

Siddhartha Sen, Manager

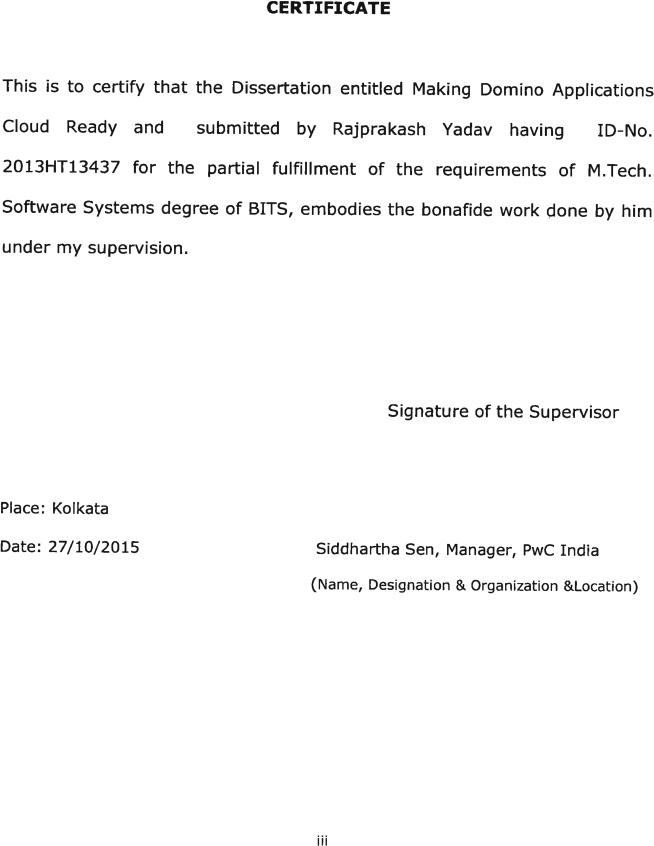
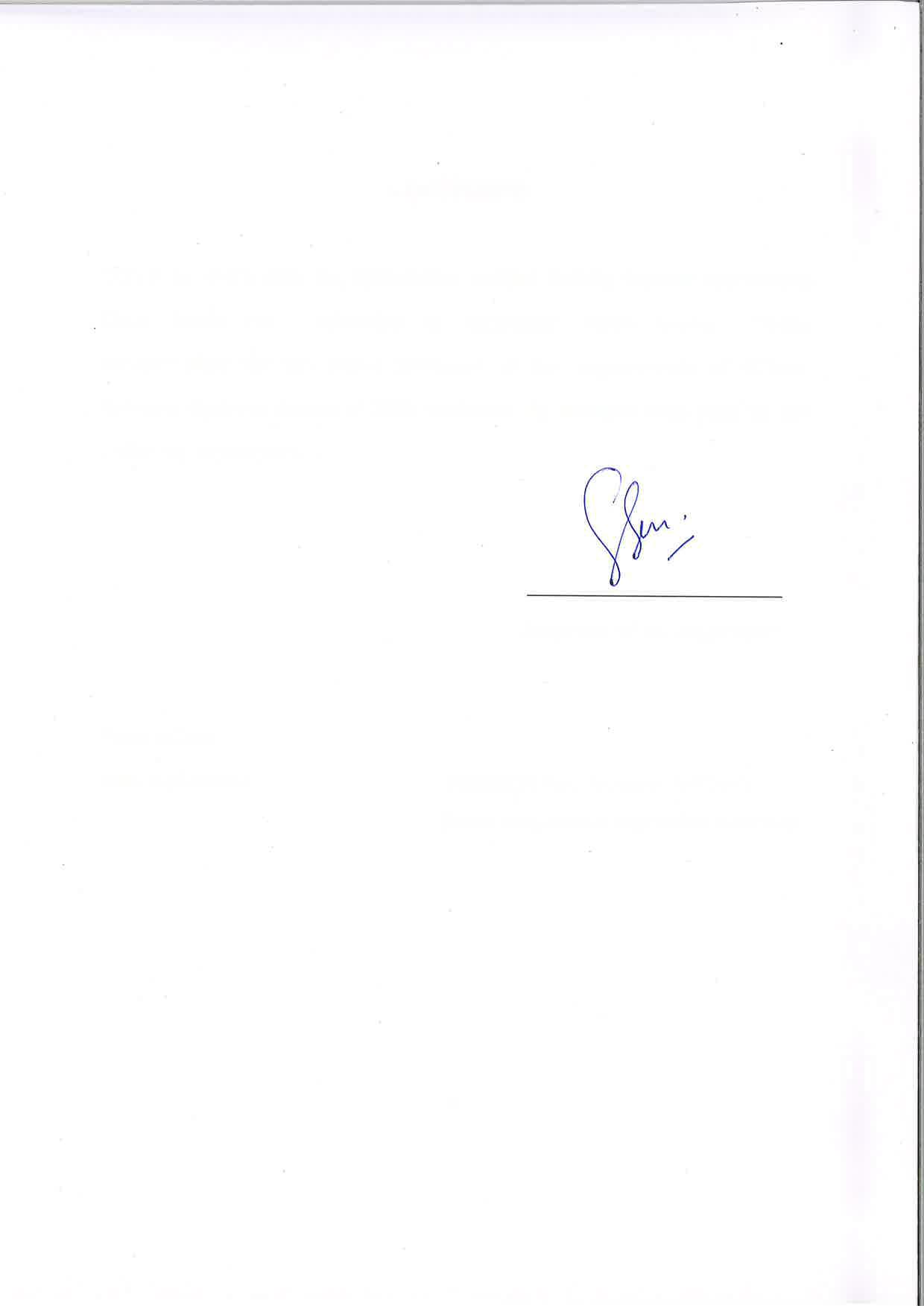
PwC India, Kolkata



**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE**

**PILANI (RAJASTHAN)**

November, 2015



**Birla Institute of Technology & Science, Pilani**

**Work-Integrated Learning Programmes Division**

**First Semester 2015-2016**

**BITS ZG628T: Dissertation**

**ABSTRACT**

**BITS ID No.** **: 2013HT13437**

**NAME OF THE STUDENT** **: RAJPRAKASH YADAV**

**EMAIL ADDRESS** **: yadav.rajprakash@gmail.com**

**STUDENT’S EMPLOYING** **: PwC India, Kolkata**

**ORGANIZATION & LOCATION**

**SUPERVISOR’S NAME** **: SIDDHARTHA SEN**

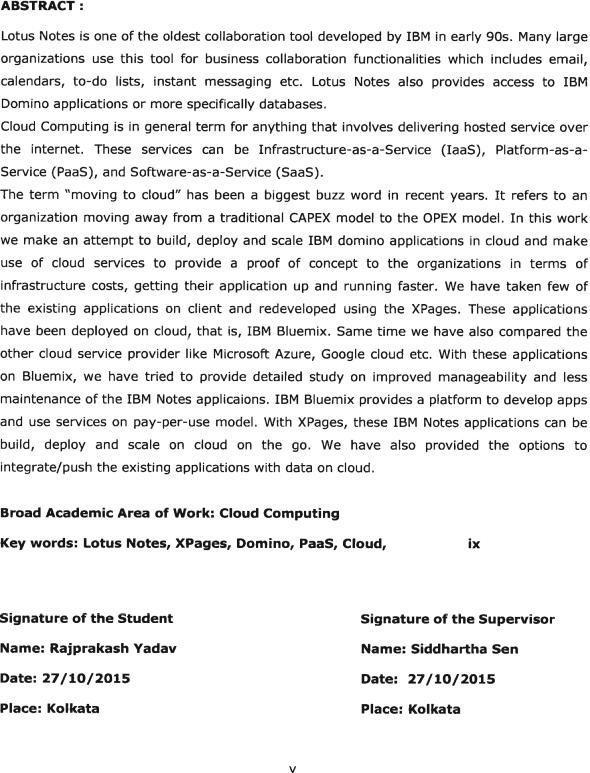
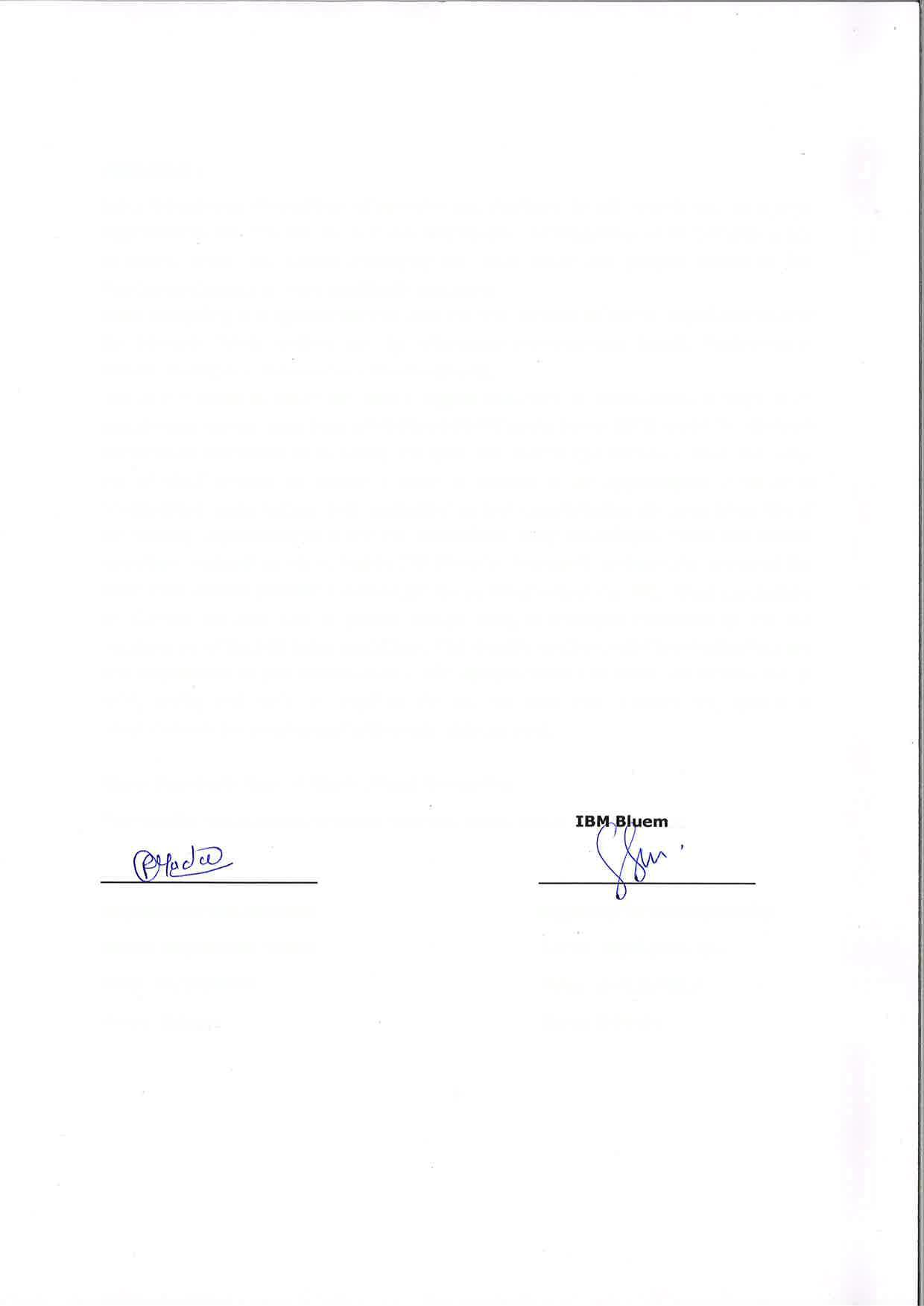
**SUPERVISOR’S EMPLOYING** **: PwC India, Kolkata**

**ORGANIZATION & LOCATION**

**SUPERVISOR’S EMAIL ADDRESS : siddhartha1.sen@in.pwc.com**

**DISSERTATION TITLE** **: Making Domino Applications Cloud Ready**

iv



**Table of Contents**

Certificate………………………………………………………………………………………………………………………………iii

Abstract…………………………………………………………………………………………………………………………………iv

Acknowledgment………………………………………………………………………………………………………………….vi

Table of Figures ix

Chapter 1: Introduction 1

1.1 IBM Notes and Domino 2

1.2 What is Cloud Computing? 2

1.3 3-4-5 Rule of Cloud Computing 4

Chapter 2: PaaS and Cloud Application Development 8

2.1 Different Cloud Service Provider 8

2.2 Windows Azure 9

2.2.1 Web Role v/s Worker Role 10

2.2.2 Azure Fabric Controller 11

2.2.3 Azure Cloud Storage Services 11

2.3 Google App Engine 12

2.3.1 Developing and deploying app on Google App Engine 12

2.4 IBM Bluemix 12

2.4.1 Bluemix Architecture 13

2.4.2 Deployment options based on Access Models 14

2.4.3 Deployment options based on Compute Infrastructure 15

2.5 Cloud Native Applications 16

2.5.1 How cloud applications are different? 16

2.5.2 Characteristics of Native Cloud Applications 16

Chapter 3: Project Overview 18

3.1 Which PaaS to use? 18

3.1.1 Application Platform Comparison 18

3.2 Why IBM Bluemix? 19

3.3 How IBM Bluemix works? 20

Chapter 4: Project Implementation 21

4.1 Application Development in IBM Domino Designer 22

4.1.1 Why XPages? 23

4.1.2 XPages Sample Code 24

4.2 Extension Library 26

Chapter 5: Domino XPages in Bluemix 27

5.1 XPages Runtime 27

5.2 Domino Data Service 27

5.3 XPages Boilerplates 28

Chapter 6: Testing and Integration 29

6.1 Creating New Applications in Bluemix 29

6.1.1 IBM Bluemix Manifest Code 30

6.2 Pushing Existing Applications to Bluemix 31

6.3 Scaling XPages Application in IBM Bluemix 34

Summary 35

Conclusion and Recommendations 36

Directions for future work 37

References 39

Checklist for the items in the report 40

**Table of Figures**

Figure 1: Overview of cloud computing 2

Figure 2: Convergence of technology fields making cloud computing 3

Figure 3: Cloud service models 4

Figure 4: SPI Model 5

Figure 5: Public Cloud 6

Figure 6: Hybrid Cloud 7

Figure 7: Different PaaS providers 9

Figure 8: Quick view of Azure Web Portal 9

Figure 9: Compute service with two Web roles and two Worker roles 10

Figure 10: Schematic diagram of Azure platform services 11

Figure 11: Bluemix Catalogue 13

Figure 12: Bluemix Architecture 14

Figure 13: Bluemix deployment options 15

Figure 14: IBM Bluemix - XPage Runtime 20

Figure 15: Deploying application on Bluemix 20

Figure 16: Invoking a Bluemix application 21

Figure 17: Basic IBM Notes Form Design 22

Figure 18: XPage Design and Source 23

Figure 19: openntf XPages Extension Library web page 26

Figure 20: XPages Runtime 27

Figure 21: Domino Data Service 27

Figure 22: XPages Boilerplates 28

Figure 23: Built-In Bluemix to Designer 29

Figure 24: Import IBM Bluemix Starter Code 30

Figure 25: NoSQL Database Service Dashboard 31

Figure 26: NoSQL Data Deployment 33

Figure 27: Bound the Services to IBM Bluemix 33

Figure 28: Sample Application on IBM Bluemix 34

Figure 29: Scaling the XPages Application 34

Figure 30: Migration Comparison based on different platforms 36

Ix

**GLOSSARY**

LEI – Lotus Enterprise Integrator

ODBC – Open Database Connectivity

NSF – Notes Storage Facility

GUI – Graphical User Interface

NIST – National Institute of Standards and Technology SOA – Service Oriented Architectures

IaaS – Infrastructure as a Service PaaS – Platform as a Service SaaS – Software as a Service

FC – Fabric Controller

Blob – Binary Large Object

REST – Representational State Transfer

API – Application Program Interface

CF CLI – Cloud Factory Command Line Interface

**Chapter 1: Introduction**

**1.1 IBM Notes and Domino**

IBM Notes and Domino [1] is an application runtime environment, email and calendaring operate as an application within IBM Notes. IBM introduced IBM Notes and Domino over 20 years ago. Since then, IBM has been providing security-rich messaging and business applications to all type of organization. IBM Notes provides business collaboration functionalities, including email, calendars, to-do lists, contacts management, team rooms, discussion forums, file sharing, microblogging, instant messaging, blogs, and user directories. IBM Notes also provides access to and integration with other IBM Domino applications and databases. IBM Notes is client and IBM Domino is server of this collaborative client-server platform.

IBM Notes and Domino consist of several components:

1. Client Applications (based on Eclipse)
2. Domino Server
3. Domino Administrative Client
4. Domino Designer

Because of application development abilities, IBM Notes and Domino compete with suites of products from other vendors such as Microsoft, Google and others. IBM Notes framework provides applications –

1. To access, store and present information through a user interface
2. To enforce security
3. To replicate

This framework stores the data in NoSQL document-database format, .nsf. The .nsf (Notes Storage Facility) file normally contains both an application design and its associated data. It can also access the other database formats i.e. relational databases and others, either through ODBC calls, LEI for Domino or web services/XPages.

As IBM Notes and Domino is an application runtime environment and build on eclipse, it provides variety of options of languages to programmers to develop applications-

* Java – directly or through XPages
* LotusScript
* Formula Language
* JavaScript

We can build applications to run either within the IBM Notes application runtime environment or through a web server for use in a web browser.

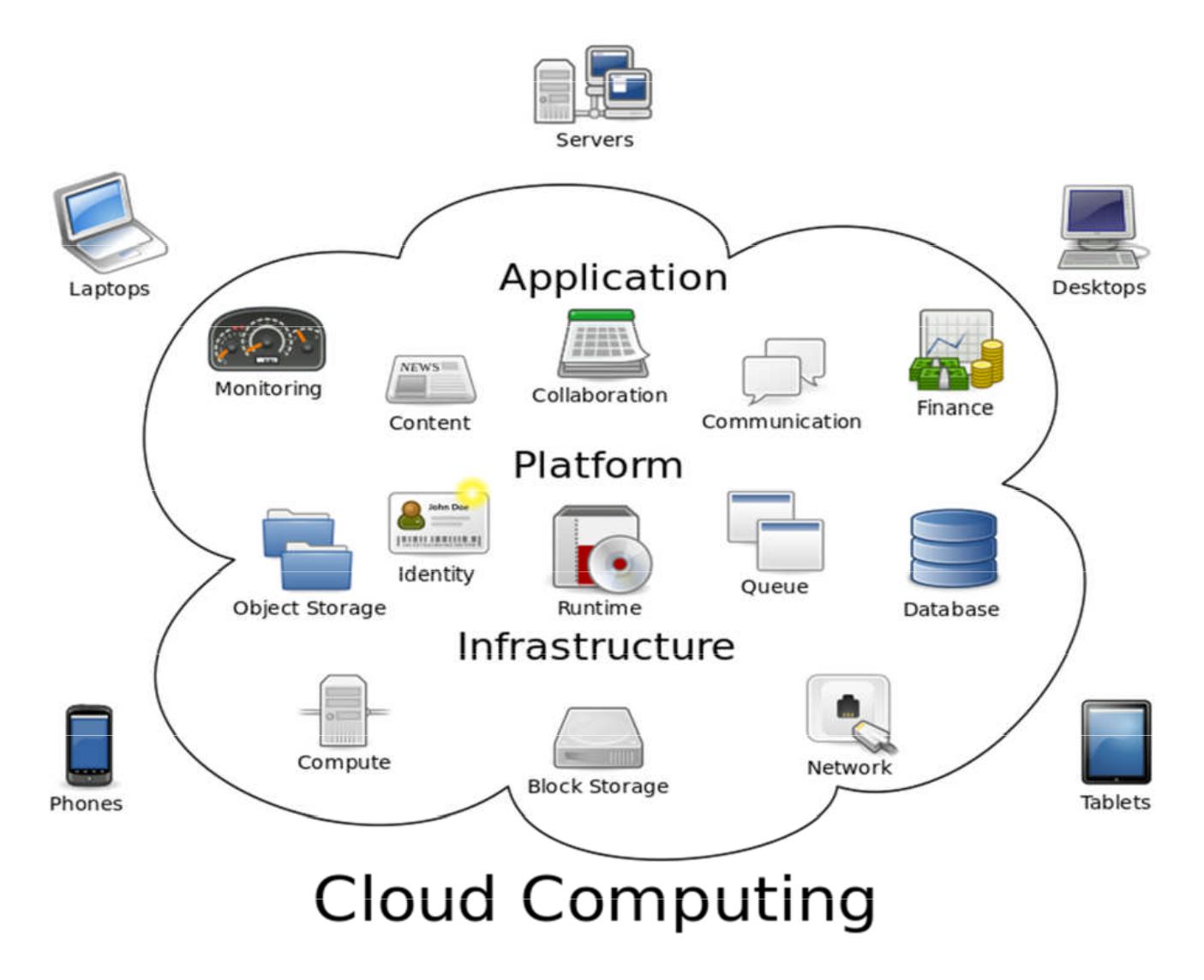
Current version of IBM Notes and Domino, IBM Notes and Domino 9 social edition delivers on the "Project Vulcan" (OneUI) vision, including the updated GUI, embedded application

**Page 1 of 40**

experiences, a significant update to iNotes to bring near-parity to the rich client, an incremental set of IBM Notes features, and the IBM Notes Browser Plug-in.

**1.2 What is Cloud Computing?**

Cloud computing [2] often referred to as simply “Cloud”. The term “Cloud” is everywhere. What is cloud? Where is cloud? Are we in the cloud now? There are all questions we will hear around us on daily basis. Cloud Computing is the delivery of on-demand computing resources—everything from applications to data centers—over the Internet on a pay-per-use basis.



**Figure 1: Overview of cloud computing**

As per Wikipedia [3] definition, Cloud Computing is –

“*a model for enabling ubiquitous, convenient, on-demand access to a shared pool of* *configurable computing resources.*”

In the simplest terms, cloud computing means storing and accessing data and programs

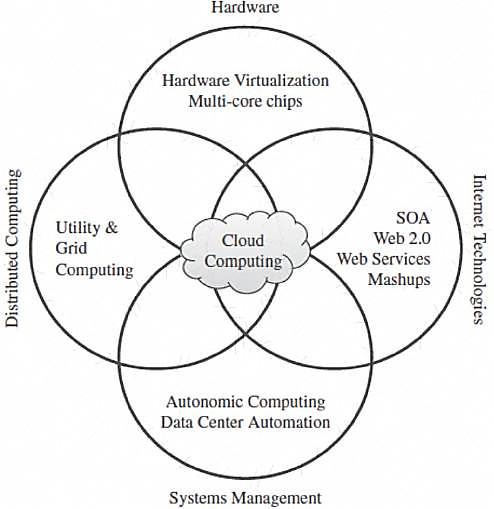
**Page 2 of 40**

over the Internet instead of computer's hard drive. The cloud is just a metaphor for the Internet. Cloud Computing is to access the data or services or programs over the internet or at least have data synchronized with other information over the Web. As an individual user, we may not have any idea what kind of massive data-processing is happening on the other end. The end result is the same - with an online connection, cloud computing can be done anywhere, anytime.

As per NIST [4], cloud computing is described as –

“Cloud computing is a model for enabling **ubiquitous**, convenient, **on-demand** network access to a **shared pool** of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be **rapidly provisioned** and released with minimal management effort or service provider interaction.”

Cloud computing [5] is like plugging an electric appliance into an outlet and get all services. It is all because of virtualization. Cloud computing is built upon hardware (that comprises of virtualization, multi-core chips etc), Internet technologies (Web Services, Service Oriented Architectures, Web 2.0), Distributed Computing (Clusters, Grids) and System Management (Autonomic computing, Data Center Automation).



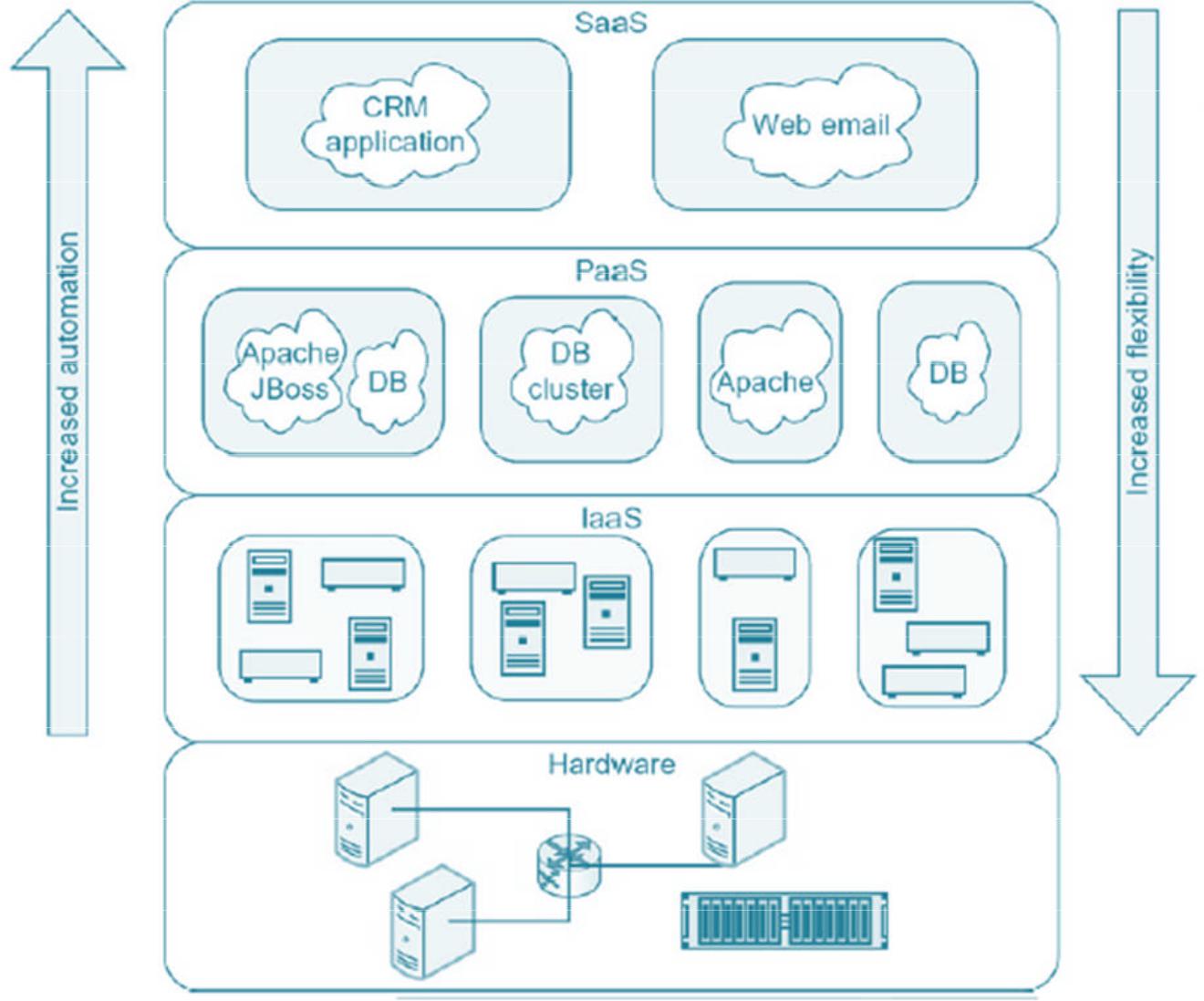
**Figure 2: Convergence of technology fields making cloud computing**

**Page 3 of 40**

**1.3 3-4-5 Rule of Cloud Computing**

1. Services
2. Deployment Models
3. Characteristics

***3: Services Provided by Cloud:*** There are 3 cloud service models [4] or service types definedby NIST for any cloud platforms. These are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).



**Figure 3: Cloud service models**

**Infrastructure as a Service:** According to NIST, IaaS [4] is defined as –“The capabilityprovided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not

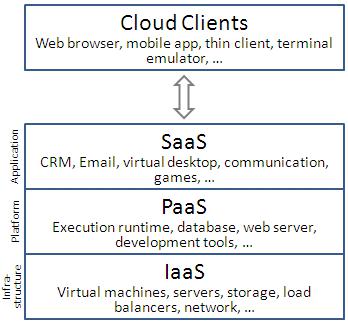
**Page 4 of 40**

manage or control the underlying cloud infrastructure but has control over operating systems; storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls)”.

**Platform as a Service:** NIST defined PaaS [4] as – “The capability provided to theconsumer is to deploy onto the cloud infrastructure consumer created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations”.

**Software as a Service:** SaaS [4] has been defined by NIST as follows: “The capabilityprovided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web based email). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user specific application configuration settings”.

The cloud service model is also called “SPI Model” by NIST [4].



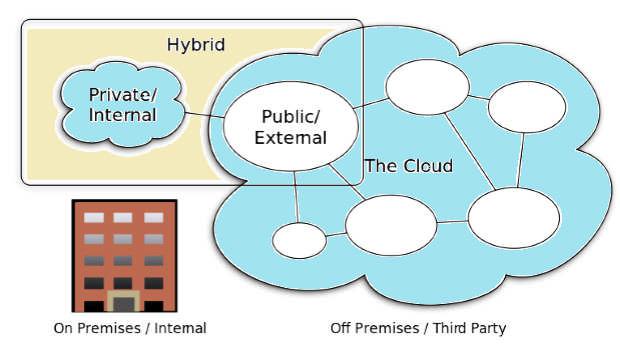
**Figure 4: SPI Model**

***4: Deployment Models:*** In addition to proposing a definition of cloud computing and 3services, NIST has defined four deployment models for cloud – Private Cloud, Public Cloud, Community Cloud and Hybrid Cloud.

**Private Cloud:** The cloud infrastructure is operated solely for a single organization. It maybe managed by the organization or a third party, and may exist on-premises or off-premises. It is also called as “Enterprise Cloud”.

**Page 5 of 40**

**Public Cloud:** The cloud infrastructure is made available to the general public or a largeindustry group and is owned by an organization selling cloud services. It is also known as “Internet Cloud”.

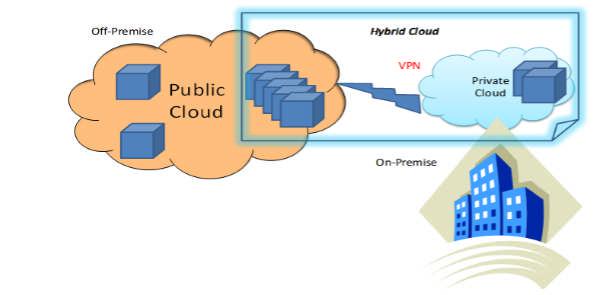


**Figure 5: Public Cloud**

**Community Cloud:** The cloud infrastructure is shared by several organizations andsupports a specific community that has shared concerns (e.g., mission, security requirements, policy, or compliance considerations). It may be managed by the organizations or a third party and may exist on-premises or off premises.

**Hybrid Cloud:** The cloud infrastructure is a composition of two or more clouds (private,community, or public). For example, leasing public cloud services when private cloud capacity is insufficient. It is also called as “Mixed Cloud”. A hybrid cloud takes shape when a private cloud is supplemented with computing capacity from public clouds. The approach of temporarily renting capacity to handle spikes in load is known as “**Cloud-bursting**”.

**Page 6 of 40**



**Figure 6: Hybrid Cloud**

***5: Characteristics:*** To further clarify the definition of cloud computing, NIST specifies thefollowing five essential characteristics [5] that a cloud computing infrastructure must have –

**On demand self-service:** The resources (compute, storage or platform) needed by theuser of a cloud platform are self-provisioned or auto-provisioned with minimal configuration.

**Broad network access:** Any computing capabilities are available over the network. Manydifferent devices are allowed access through standardized mechanisms.

**Resource pooling:** Multiple users can access clouds that serve other consumers accordingto demand. Cloud services need to share resources between users and clients in order to reduce costs.

**Rapid elasticity:** Provisioning is rapid and scales out or in based on need.

**Measured or metered service:** One of the compelling business use cases for cloudcomputing is the ability to "pay as you go“, where the consumer pays only for the resources that are actually used by his applications.

**Page 7 of 40**

**Chapter 2: PaaS and Cloud Application Development**

Platform as a Service (PaaS) model provides a platform on which users can directly develop and deploy their applications without worrying about the complexity of setting up the hardware or system software. Should application developers be burdened by tasks of ensuring that a specific server is up and running? Should they worry about the disk space? Should they worry about which OS their app should support? The focus should be on solving the big problems. The compute infrastructure, platform, libraries and application deployment should all be automated and abstracted. This is where Cloud Computing i.e. PaaS plays a major role.

The PaaS model provides the tools within an environment needed to create applications that can run in Software as a Service model. PaaS is application middleware offered as a service to developers, integrators, and architects. Development and Operation teams use PaaS to design, build, and deliver customized applications or information services. Instead of relying on standardized SaaS, teams using PaaS have more control over solution architecture, quality of service, user experience, data models, identity, integration, and business logic. In PaaS you are given a toolkit to work with, a virtual machine to run your software on and it is up to you to design the software and its user-facing interface in a way that is appropriate to your needs. So PaaS systems range from full-blown developer platforms like Windows Azure to systems like Drupal, Squarespace, Wolf, and others where the tools are modules that are very well developed and require almost no coding. PaaS solution will ensure the availability of the application despite downtime of the underlying virtual machine by automatically creating a new instance of the application on a new virtual machine when the machine goes down. PaaS systems can be used to host a variety of cloud services –

Online portal-based applications like Facebook that need to scale to thousands of Users

Startup who wants to host their new application in a Software-as-a-Service model Can also be used for massively parallel computations

Enterprises can deploy their Line-of-Business applications in the cloud, taking advantage of the scale and availability while still maintaining security and privacy of data

1. **Different Cloud Service Provider**

In this section, we will have a quick look on different cloud service provider especially for PaaS model. There are numerous PaaS service provider in the current market. Below picture captures few of them –

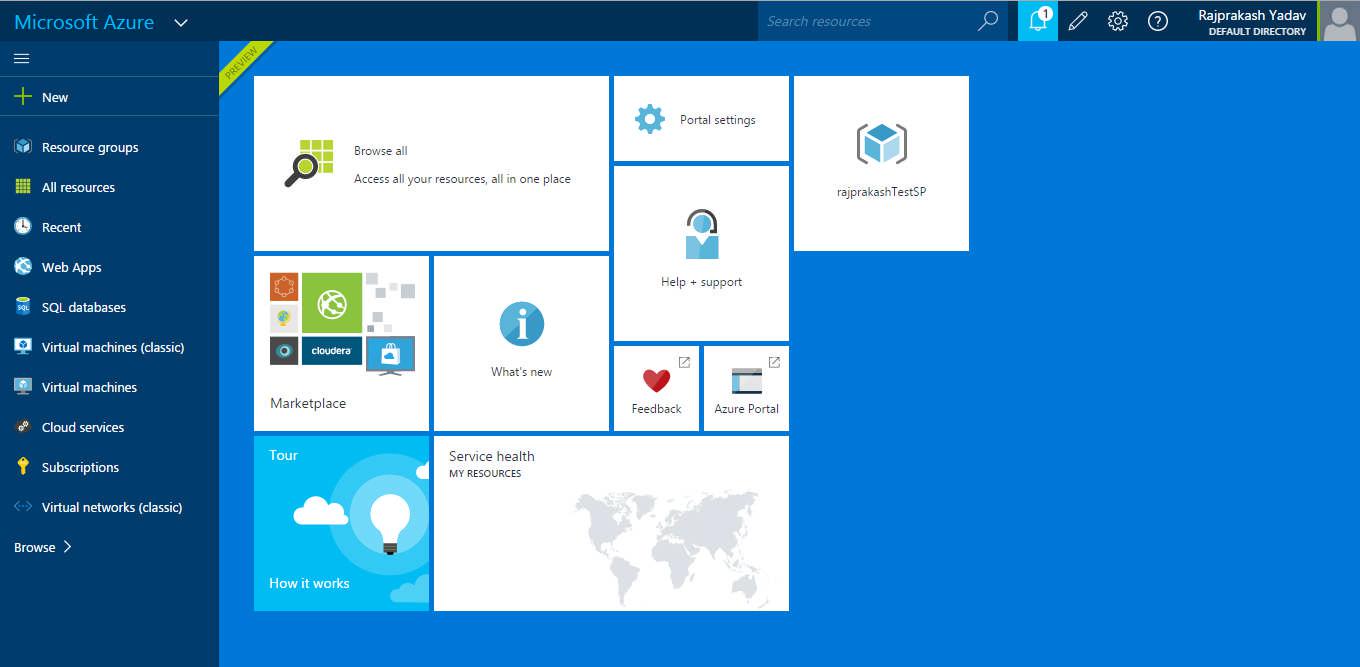
**Page 8 of 40**



**Figure 7: Different PaaS providers**

**2.2 Windows Azure**

The Azure Services Platform [6] is a popular application platform for the cloud that allows Windows applications and web-services to be hosted and run in Microsoft datacenters. It is cloud deployment platform for applications developed for Windows using .NET.



**Figure 8: Quick view of Azure Web Portal**

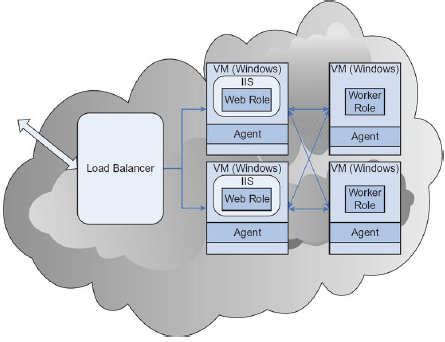
**Page 9 of 40**

While Azure is primarily designed for PaaS capabilities, it also includes certain features for Data-as-a-Service (DaaS) and Infrastructure-as-a-Service (IaaS). Like any other cloud-hosted application, Windows Azure applications typically have multiple instances running on different virtual machines. Developers need not create nor manage these virtual machines explicitly. Developers just write applications either as a Web role and/or Worker role, and tell Windows Azure the number of instances of each role that should be created.

**2.2.1 Web Role v/s Worker Role**

Applications are commonly composed of both Web and Worker roles. We can think of Compute as being a container for web roles and worker roles. Web roles are simply front-end web applications and content hosted inside of IIS in a Microsoft data center. We can quickly and easily deploy web applications to Web Roles and then scale our Compute capabilities up or down to meet demand. Worker roles allow us to host any type of application, including Apache Tomcat and Java Virtual Machines (JVM). A common implementation in Windows Azure takes input from a Web role, sends those requests through a Queue to a Worker role, then processes the requests and stores the output.

Worker roles are used for the more heavy-duty processing that is typically done by Windows services in on premise applications. Web roles and Worker roles communicate by either using message queues or by setting up direct connections via WCF or other technologies.

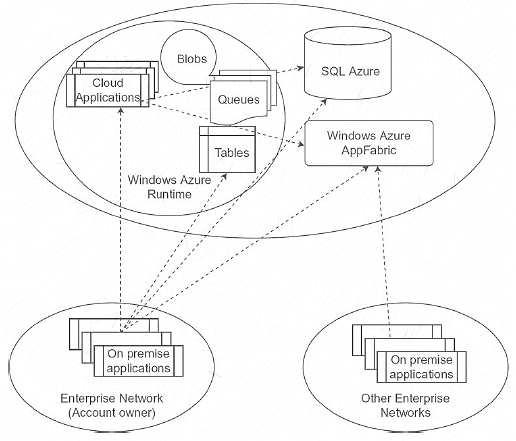


**Figure 9: Compute service with two Web roles and two Worker roles**

Microsoft provides Visual Studio project templates for creating Windows Azure Web roles, Worker roles, and combinations of the two, and developers are free to use any programming language supported on Windows. Eclipse for Java and PHP development is

**Page 10 of 40**

also supported.



**Figure 10: Schematic diagram of Azure platform services**

**2.2.2 Azure Fabric Controller**

Windows Azure runs on a large number of machines, all maintained in Microsoft data centers. The hosting environment of Azure is called the Fabric Controller (FC) [5] [6]. The FC is a distributed program that manages the hardware and applications in a cluster internally used by Azure. The key task of FC is to assign the appropriate resources to an application.

**2.2.3 Azure Cloud Storage Services**

There are four storage services [7] under windows azure –

1. **Azure Blob Services:** Blobs is Binary Large Objects. It provides file-levelstorage. Blobs are always stored under containers. Every storage account must have at least one container, and containers can have blobs within them. Container names can contain the directory separator character ("/"). This gives developers the facility to create hierarchical "file-systems". The blob service defines 2 kinds of blobs to store text and binary data: A page blob and a block blob.
2. **Azure Files/Drives:** SMB based Azure Files for migrating on-premisesapplications to the cloud. Windows Azure Drives are used for mounting an NTFS volume to be accessed by an application.
3. **Azure Table Services:** Structured noSQL forms of storage using key-valuepairs. Table Storage can store petabytes of semi-structured data while keeping

costs

**Page 11 of 40**

down. Using geo-redundant storage, stored data is replicated 3 times within a region—and an additional 3 times in another region more than hundreds of miles apart.

* 1. **Azure Queues:** Provide reliable message delivery within and between services. Astorage account can have unlimited number of queues, and each queue can store an unlimited number of messages. Queues are used by Web roles and Worker roles for inter-application communication, and by applications to communicate with each other.

1. **Google App Engine**

Google App Engine [8] is a PaaS solution that enables users to host their own applications on the Google data centers similar to Google Docs, Google Maps and other popular Google services. It enables users to develop and host applications written using Java, Python, Go, JRuby, JavaScript (Rhino), Scala etc. The applications hosted on Google App Engine can scale both in compute and storage just like other Google products.

**2.3.1 Developing and deploying app on Google App Engine**

Google App Engine supports apps written in a variety of programming languages. Applications run in a secure, sandboxed environment, allowing App Engine to distribute requests across multiple servers, and scaling servers to meet traffic demands. Software Development Kits (SDKs) for App Engine are available in all supported languages. Each SDK includes:

All of the APIs and libraries available to App Engine.

A simulated, secure sandbox environment, that emulates all of the App Engine services on your local computer.

Deployment tools that allow you to upload your application to the cloud and manage different versions of your application.

**To get started on GAE:**

* 1. Download and install Java EE
  2. Add plug-ins: Google plugin for eclipse, SDK
  3. Create a new "Web Application Project"
  4. Configure the application
  5. Develop code
  6. Test in simulated App Engine environment
  7. Deploy to Google App Engine

1. **IBM Bluemix**

IBM Bluemix [9] is the cloud platform that helps developers rapidly build, manage and run web and mobile applications. Bluemix is the latest cloud offering from IBM. It enables organizations and developers to quickly and easily create, deploy, and manage applications on the cloud. Bluemix is an implementation of IBM's Open Cloud Architecture based on Cloud Foundry, an open source Platform as a Service (PaaS).

**Page 12 of 40**

**What is Cloud Foundry?**

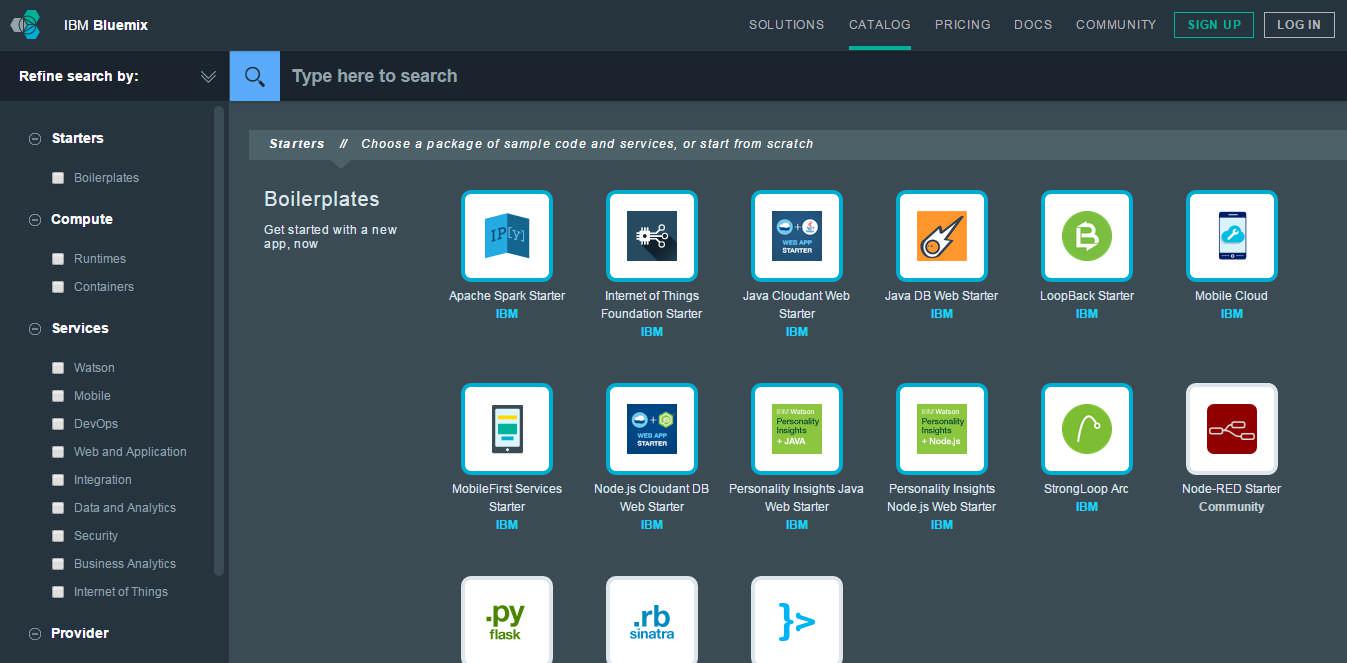
Cloud Foundry [10] is an open source platform as a service (PaaS) that let us quickly create and deploy applications on the cloud. It provides different frameworks and services.

**Development Framework:** Java, Spring, Ruby, Node.js and custom frameworks.

**Application Services:** Support for MySQL, MongoDB, PostgreSQL, Redis, RabbitMQ etc.

**Clouds:** Cloud Foundry in Public, Private, VMWare and OpenStack-based clouds.

In addition to providing frameworks and services based on Cloud Foundry, Bluemix provides a dashboard for us to create, view, and manage our applications. Bluemix dashboard provides services to monitor our application’s resource usages and also provides the ability to manage organizations, spaces, and user access.



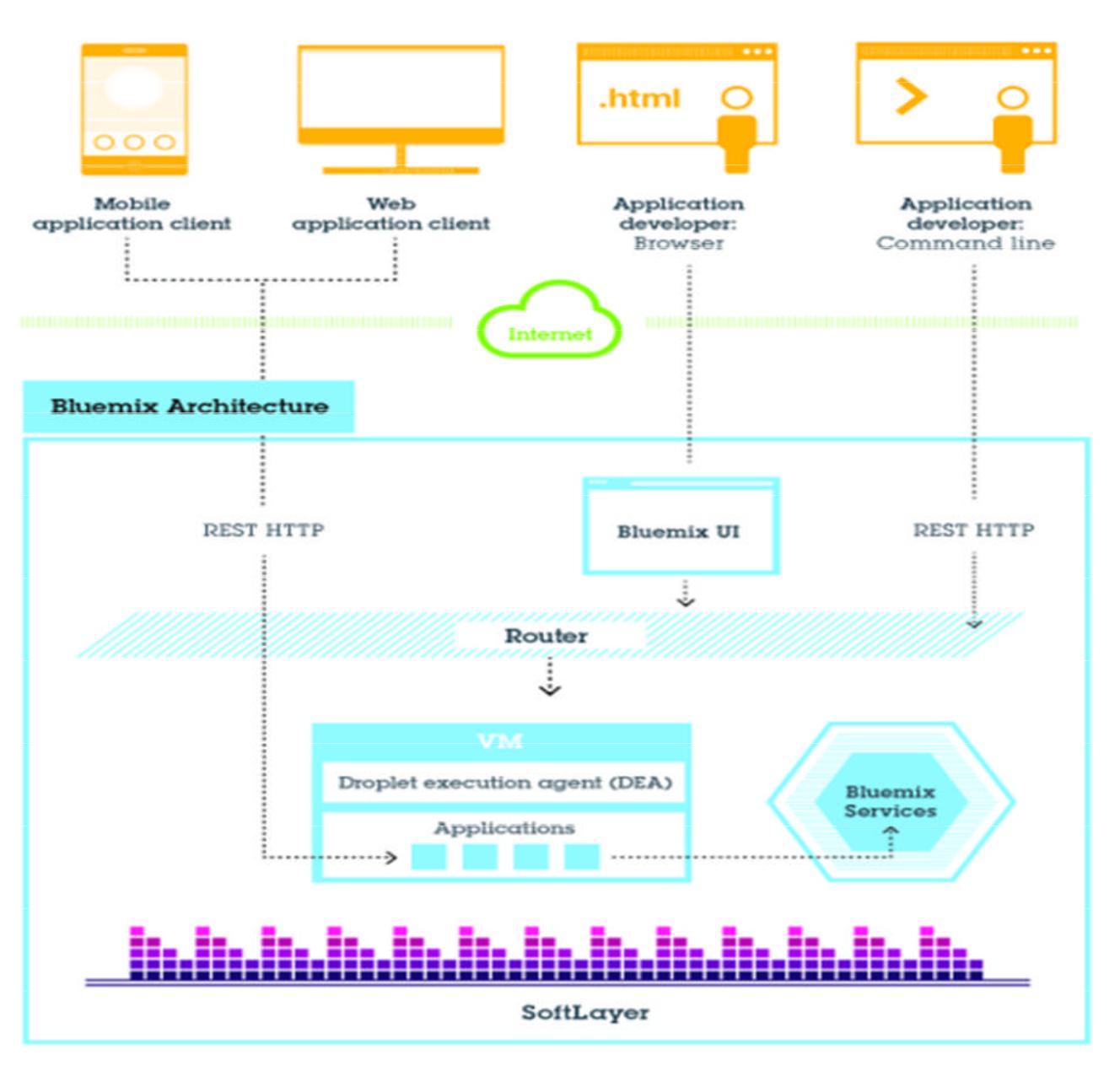
**Figure 11: Bluemix Catalogue**

**2.4.1 Bluemix Architecture**

Bluemix is a platform to develop apps and use services that provide ready-to-use functions. By using SoftLayer [9], Bluemix deploys virtual containers that host each deployed app. The app can use pre-built services (including third-party services) to make app assembly easy.

With Bluemix, we can access the public Bluemix platform, set up a dedicated Bluemix platform, or use both. As a developer, we can interact with the Bluemix infrastructure by using a browser-based user interface. We can also use a Cloud Foundry command line interface, called cf, to deploy web apps.

**Page 13 of 40**



**Figure 12: Bluemix Architecture**

**2.4.2 Deployment options based on Access Models**

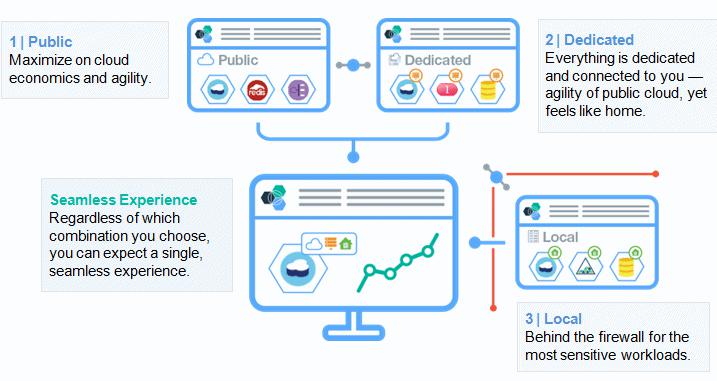
Bluemix allows choosing a deployment model based on location or privacy requirements of application and data, shared or private environment and level of infrastructure abstraction needed.

**Public:** Bluemix is offered as a public platform as a service (PaaS) environment leveragingSoftLayer virtual servers to anyone over the internet. This model allows a developer to just make an account and start developing and deploying applications.

**Dedicated:** Bluemix Dedicated is exclusive SoftLayer environment that’s securelyconnected to both the public Bluemix and network. Bluemix Dedicated is connected to network through a VPN or a direct network connection. IBM manages the dedicated platform

**Page 14 of 40**

and dedicated services, so that we can focus on building custom applications.



**Figure 13: Bluemix deployment options**

**Local:** Bluemix Local is for enterprises that want to have their own environments and bringsall the advantages of a PaaS into their private world and own data center. It is delivered as a fully managed service in organization’s own data center and brings cloud agility and Bluemix features to even the most sensitive workloads. Bluemix Local sits either on OpenStack or VMWare driven infrastructure, or on a Bluemix appliance.

**2.4.3 Deployment options based on Compute Infrastructure**

Similar to the access models, there are 3 deployment options for Bluemix.

1. **Cloud Foundry:** IBM Bluemix is based on Cloud Foundry - an open PaaS (platformas a service). In this PaaS model, Bluemix takes care of the management and maintenance of the all the infrastructure and runtime that powers our application. This enables developers to build applications rapidly in local IDEs and push the applications to the cloud.
2. **IBM Containers:** With this option, we can run our application code in an IBMcontainer that is based on the popular Docker containers. We can port our applications more easily since we can run these containers locally, on-premise environments and on cloud.
3. **Virtual Machines:** In this option, we can run our application code on our ownVirtual machine with its own Operating System.

**Page 15 of 40**

**2.5 Cloud Native Applications**

Cloud native applications [11] are developed specifically for cloud platforms. Being cloud-native is more about the application architecture and design than how we code the thing. To properly take advantage of a cloud platform, including IaaS and PaaS, we have to design the applications so that they're decoupled from any specific physical resource.

When this architecture is considered in the design, development, and deployment of an application, the utilization of the underlying cloud resources can be more efficient. We are paying for the resources we use, so applications that more efficiently work with those resources run faster and generate smaller cloud services bills at the end of the month.

**2.5.1 How cloud applications are different?**

The cloud was originally marketed [12] to developers as “the same thing we have today but a vendor is taking care of the infrastructure”. In some ways this is true, take any cloud vendor that allows us to request a VM from them. It feels just like the VM we had running in our own data center for the most part. The only difference is that it’s NOT IN OUR DATA CENTER. The fact that it is not running in our datacenter is exactly the reason why cloud applications need to be architected differently.

**2.5.2 Characteristics of Native Cloud Applications**

As mentioned in Matt Stine’s book [13] “Migrating to Cloud – Native Application Architectures”, cloud native applications should have following characteristics –

Is a 12-factor application

Follows as microservice architecture Uses PaaS

Uses API based collaboration Is antifragile

***12-Factor Application***

With Heroku [14], engineers came up with different set of guidelines for developers to use the platform. There are 12 common practices [15] listed below –

1. **Codebase:** One codebase tracked in revision control, many deploys
2. **Dependencies:** Explicitly declare and isolate dependencies
3. **Config:** Store config in the environment
4. **Backing Services:** Treat backing resources as attaches resources
5. **Build, release, run:** Separate build and run stages
6. **Processes:** Execute the app as one or more stateless resources
7. **Port Binding:** Export services via port binding
8. **Concurrency:** Scale out via the process model
9. **Disposability:** Maximize robustness with fast startup and graceful shutdown **10.Dev/Prod Parity:** Keep development, staging, and production as similar as possible **11.Logs:** Treat logs as events streams

**12.Admin Processes:** Run admin/management tasks as one-off processes

**Page 16 of 40**

***Microservice Architecture***

Microservices is an architecture style [13], in which large complex software applications are composed of one or more services. Microservice can be deployed independently of one another and are loosely coupled. Each of these microservices focuses on completing one task only and does that one task really well. That one task represents a small business capability. Microservices can be developed in any programming language. They communicate with each other using language-neutral application programming interfaces (APIs) such as Representational State Transfer (REST) [21]. They don’t need to know anything about underlying implementation or architecture of other microservices.

***Self-Service Agile Infrastructure***

Self-service agile infrastructure [11] is nothing but PaaS. When deploying cloud native applications we do not want to go through the process of requesting a VM, configuring that VM, securing that VM, installing dependencies, deploying our application, configuring our application, and setting up tools to monitor our application, we want a platform to do that for us. If our cloud native application is using microservices it can potentially have 100s of services that need to be deployed in order for the application to work. Going through the above process for each microservice is not going to scale, no matter how many people work on the team.

The self-service agile infrastructure should be responsible for providing our application with the backing services which needs to run. Things like databases, message queues, and caching services should be able to request on demand and provisioned rapidly when an application is deployed.

***API Based Collaboration***

When we are following the microservice design pattern we are going to need a new way for the components in our application to communicate. We cannot rely on just making method calls between the components of our application anymore. REST APIs are the most common way for services to communicate in cloud native applications. The most robust cloud native applications today support desktops, browsers, phones, tablets, watches, TVs, cars, entertainment systems, game consoles, and may other types of clients. The protocol that works best for all these types of devices is REST.

***Antifragile***

Today’s cloud native applications are expected to have 0 down time. A crash in one component should still leave the majority of our application available to our users, only if we are using microservice design pattern. If we take advantage of different data centers and regions from our cloud provider we have enough redundancy to withstand an entire outage of our application in one region without anyone even noticing.

**Page 17 of 40**

**Chapter 3: Project Overview**

More and more users of the IBM Lotus Notes and Lotus Domino systems today are considering migrating to other systems. Users considering the migration of their Notes client applications to other platforms to-

eliminate the dependency on the Notes client web enable or mobile enable the applications enhance usability or

lower maintenance/licensing costs

The advent of cloud computing makes this migration much easier to consider. For Lotus Notes/Domino users, cloud-based, hosted messaging and application services can provide the latest technologies along with an attractive alternative to costly upgrades. But if organization is considering migrating to the cloud, we should carefully examine which hosted services can most easily inherit existing technology investments, and identify tradeoffs that might adversely impact IT teams and end users.

**3.1 Which PaaS to use?**

There are many cloud options in the market. Migration to any new environment will almost always impact end users, but we can minimize that impact by choosing the right cloud/hosted service provider.

Generally document repositories and forms-based business applications run on IBM Lotus Notes/Domino servers. To transition these applications to the cloud, capabilities such as built-in workflow, role-based security and business logic must be redeveloped and supported. Google’s strategy document [16] for transitioning IBM Lotus Notes applications states that if an organization’s “migration costs don’t justify an appropriate return on investment, or users and business processes depend on a small set of Notes applications in a way that complicates migration, then continuing to support a Lotus Notes installation may be the logical path.”

If organizations want to transition away from IBM Lotus Notes completely, they must consider the custom applications running in their environment. Many organizations that have moved their email to Exchange Online have hesitated to move their Lotus Notes applications to SharePoint.

**3.1.1 Application Platform Comparison**

Both Microsoft and Google offer application-development environments as target platforms for IBM Lotus Notes applications. But only Microsoft allows deploying applications both on premises and in the cloud. We can deploy SharePoint business applications in the cloud, on premises, or in a hybrid manner—both in the cloud and on-premises. SharePoint 2013 [16] offers development capabilities that require no coding, and advanced content-management features such as Managed Metadata to address the business needs of customers.

By comparison, Google does not provide an on-premises option for deploying business applications. All applications must be redeveloped in their hosted services environment and

**Page 18 of 40**

supported in the cloud. If we cannot re-platform our IBM Lotus Notes applications to Google’s hosted services environment due to redevelopment costs, security concerns, or data-management issues, then we will likely need to retain our IBM Lotus Notes environment.

Google’s version of an application development platform is a mixture of solutions, including Google Sites, Google Apps Scripts, Google Spreadsheets, Google Gadgets, and Google App Engine. For custom development, we must rebuild all IBM Lotus Notes business applications from scratch, using HTML for forms, plus standard Java technologies, including the JVM, Java servlets, and the Java programming language—or any other language using a JVM-based interpreter or compiler, such as JavaScript or Ruby.

**3.2 Why IBM Bluemix?**

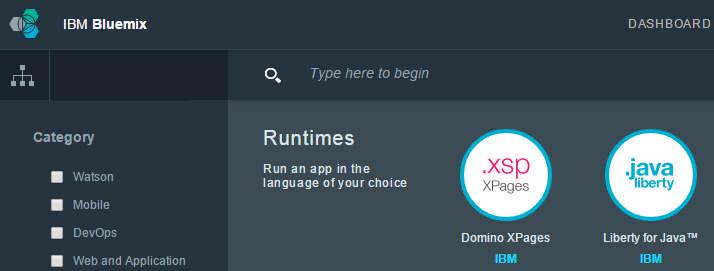
Redeveloping a Lotus Notes application requires redesigning the forms for creating and editing data documents. Transitioning entire applications of IBM Lotus Notes items to the cloud or an on-premises platform often requires rebuilding highly customized, mission-critical business applications and migrating data to the target platform. IBM Notes and Domino platform migration of Notes client applications is an expensive and labor intensive task that poses a real and significant risk of business disruption. This risk increases significantly with the number and complexity of Notes applications being migrated. In addition to the almost impossible task of replicating the many IBM Domino capabilities, platform migration requires data migration which increases the number of applications that should be considered for migration to approximately 20%-30% of the total number of Notes applications.

Instead, IBM Domino and Notes customers should consider enabling their Notes applications in XPages applications. This eliminates the need for data migration and will reduce the number of applications that should be considered for enablement to approximately 5%-10% while providing the same goals of a platform migration for a fraction of the cost, effort, resources and risk.

At IBM ConnectED 2015 [17] IBM announced a new capability which allows developers to run XPages applications on IBM Bluemix. A new XPages runtime, a new Domino data service and a new XPages boilerplate were introduced. With this introduction of XPages on cloud, it is clearly best place to host IBM Notes application. IBM Bluemix currently provides this capability under experimental lab services and we hope to see a fully-fledged support for XPages on this PaaS platform.

With support of XPages applications running on Bluemix, IBM provides new options for developers to host XPages applications in the cloud. In addition to other options where Domino is hosted on Infrastructure as Service offerings like Softlayer, developers can also easily use other services in Bluemix and mix those in their applications. For example XPages applications can use the cognitive Watson services in Bluemix, XPages applications can store data in other databases without having to set up infrastructure first, they can leverage the auto-scalability functionality and data caching of Bluemix and much more.

**Page 19 of 40**



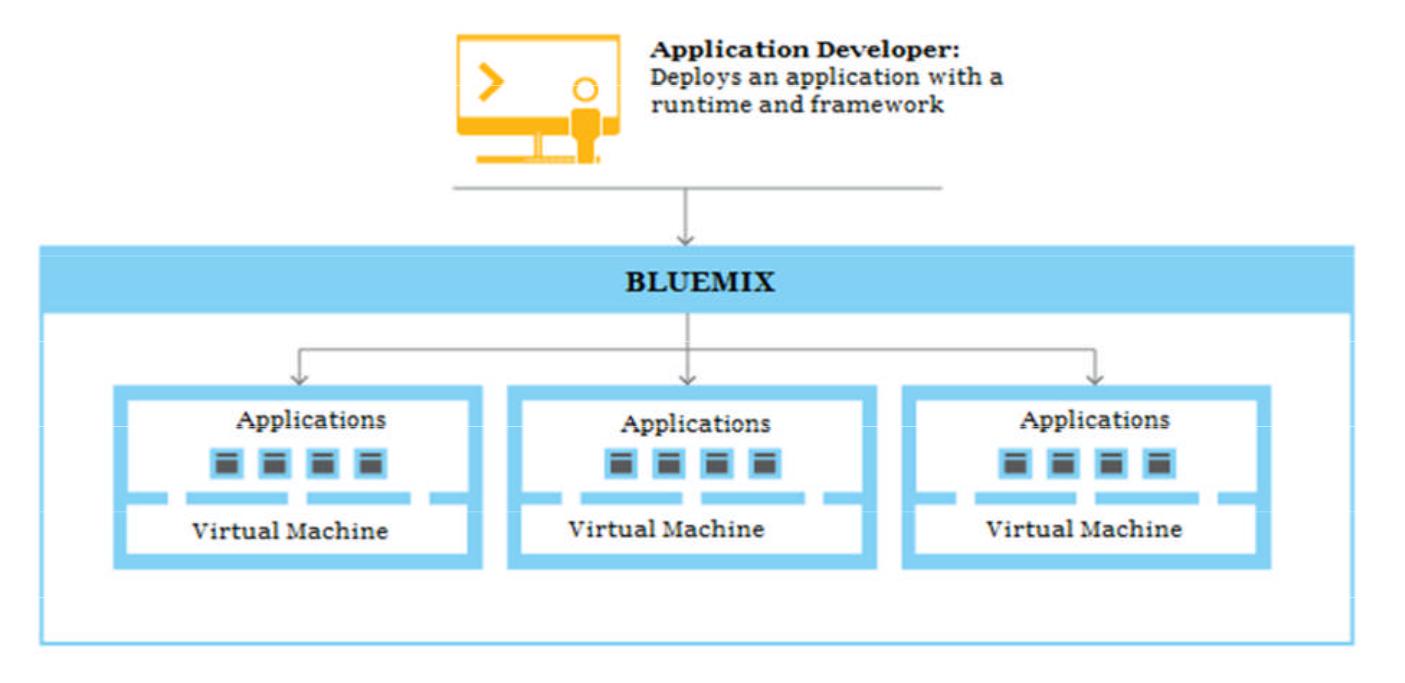
**Figure 14: IBM Bluemix - XPage Runtime**

**3.3 How IBM Bluemix works?**

When we deploy an app to Bluemix, we must configure [17] Bluemix with enough information to support the app.

For a mobile app, Bluemix contains an artifact that represents the mobile app's back end, such as the services that are used by the mobile app to communicate with a server.

For a web app, we must ensure that information about the proper runtime and framework is communicated to Bluemix, so that it can set up the proper execution environment to run the app.



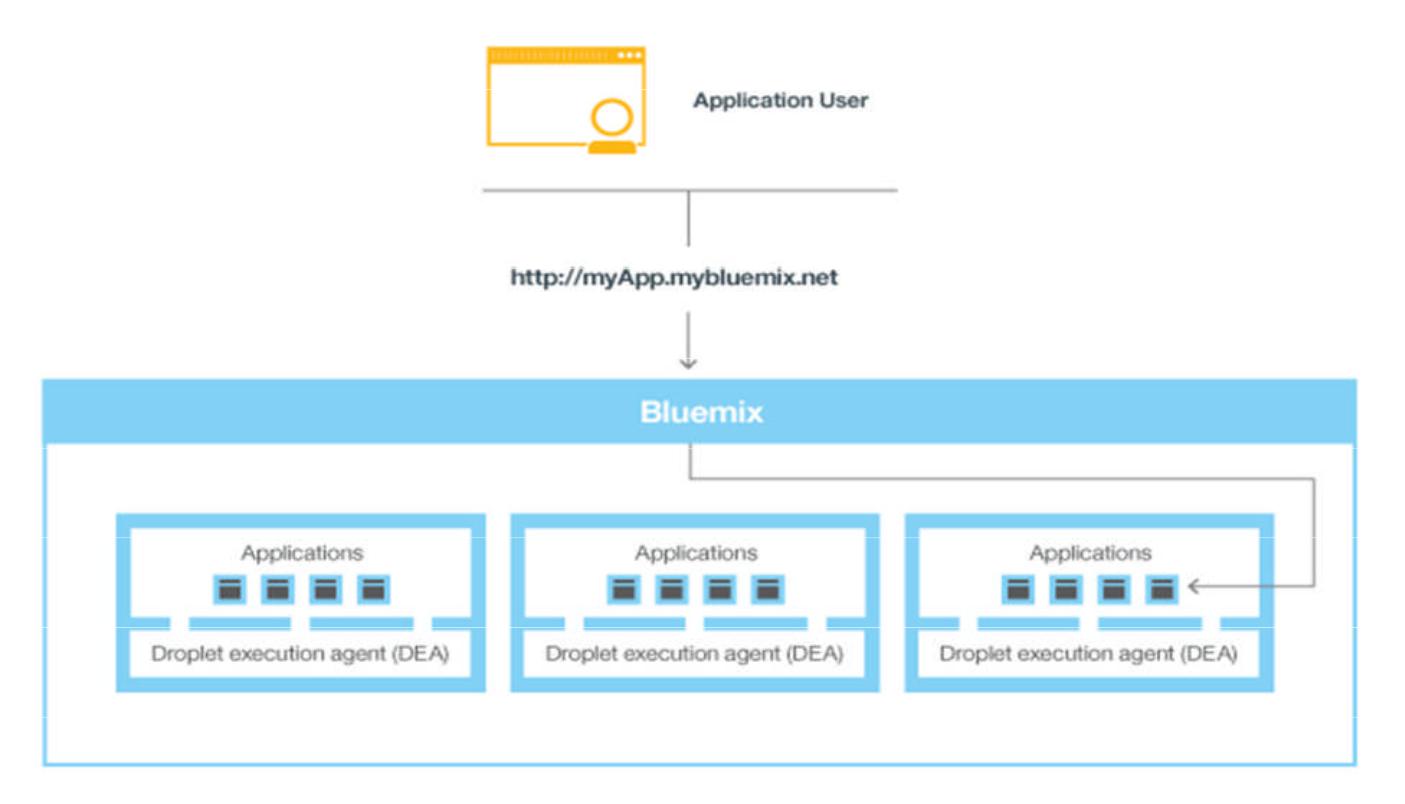
**Figure 15: Deploying application on Bluemix**

**Page 20 of 40**

When we create an app and deploy it to Bluemix, the Bluemix environment determines an appropriate virtual machine (VM) to which the app or artifacts that the app represents is sent. After a VM is chosen, an application manager on each VM installs the proper framework and runtime for the app. Then the app can be deployed into that framework. When the deployment is completed, the application artifacts are started.

In each VM, an application manager communicates with the rest of the Bluemix infrastructure, and manages the apps that are deployed to this VM. 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 all of them must point to the Bluemix endpoint. When a request comes in, Bluemix examines the request, determines which app it is intended for, and then selects one of the instances of the app to receive the request.



**Figure 16: Invoking a Bluemix application**

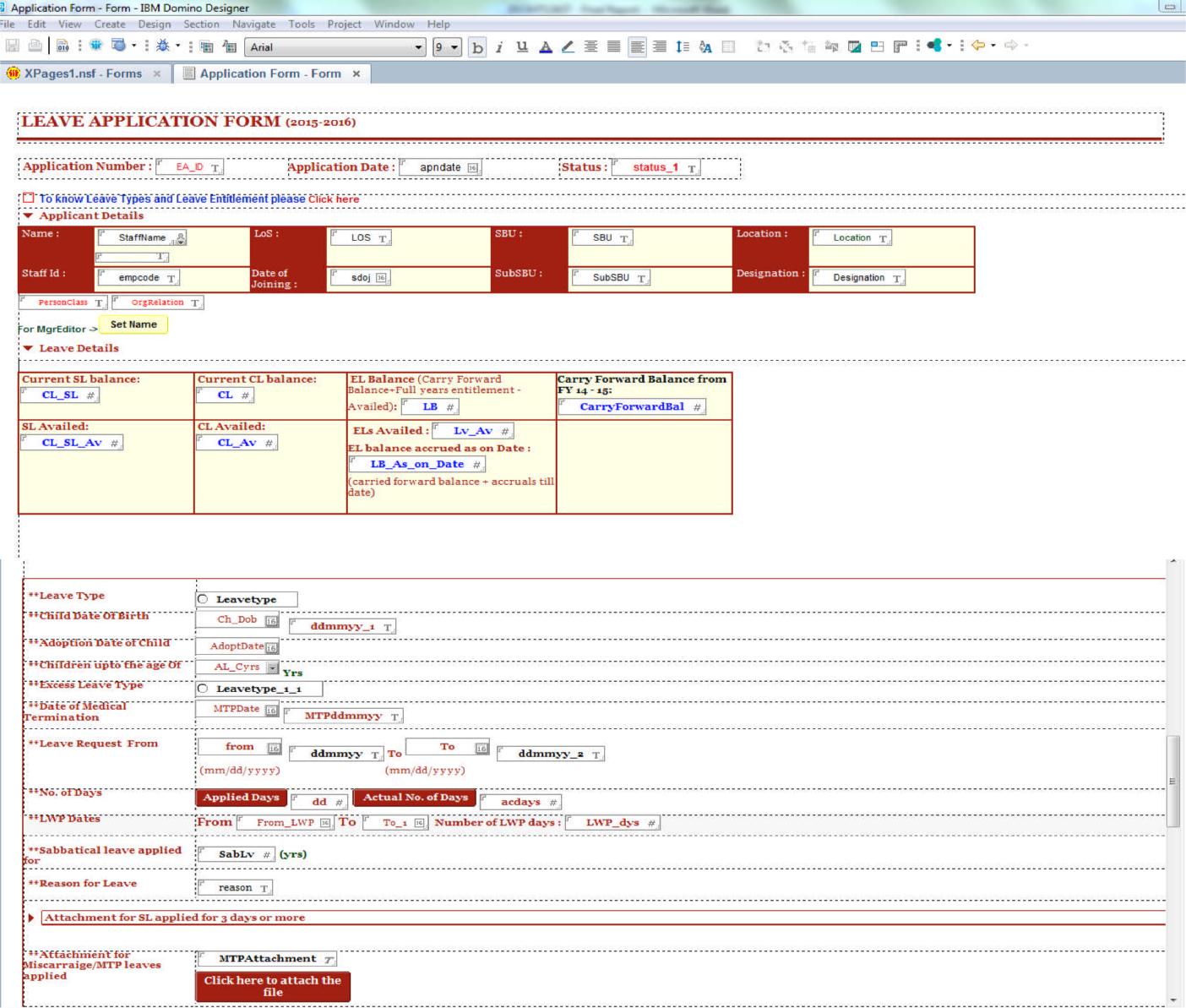
**Page 21 of 40**

**Chapter 4: Project Implementation**

In this work, our main focus is on building, deploying and scaling the IBM Notes application on cloud. We have used an existing IBM Notes application, Leave Management System, which is designed to support only Notes client. This is widely used application across the big organization and we have proposed to push this particular application on cloud as there are no such private data on this application and can be pushed even on public cloud.

**4.1 Application Development in IBM Domino Designer**

As this work is on existing IBM Notes application, we have not done any work on Notes client side. Notes legacy applications generally designed based on the formula language and lotusscript. Form based notes applications have fields and action buttons. Basically workflow is designed based on buttons embedded on the form itself.

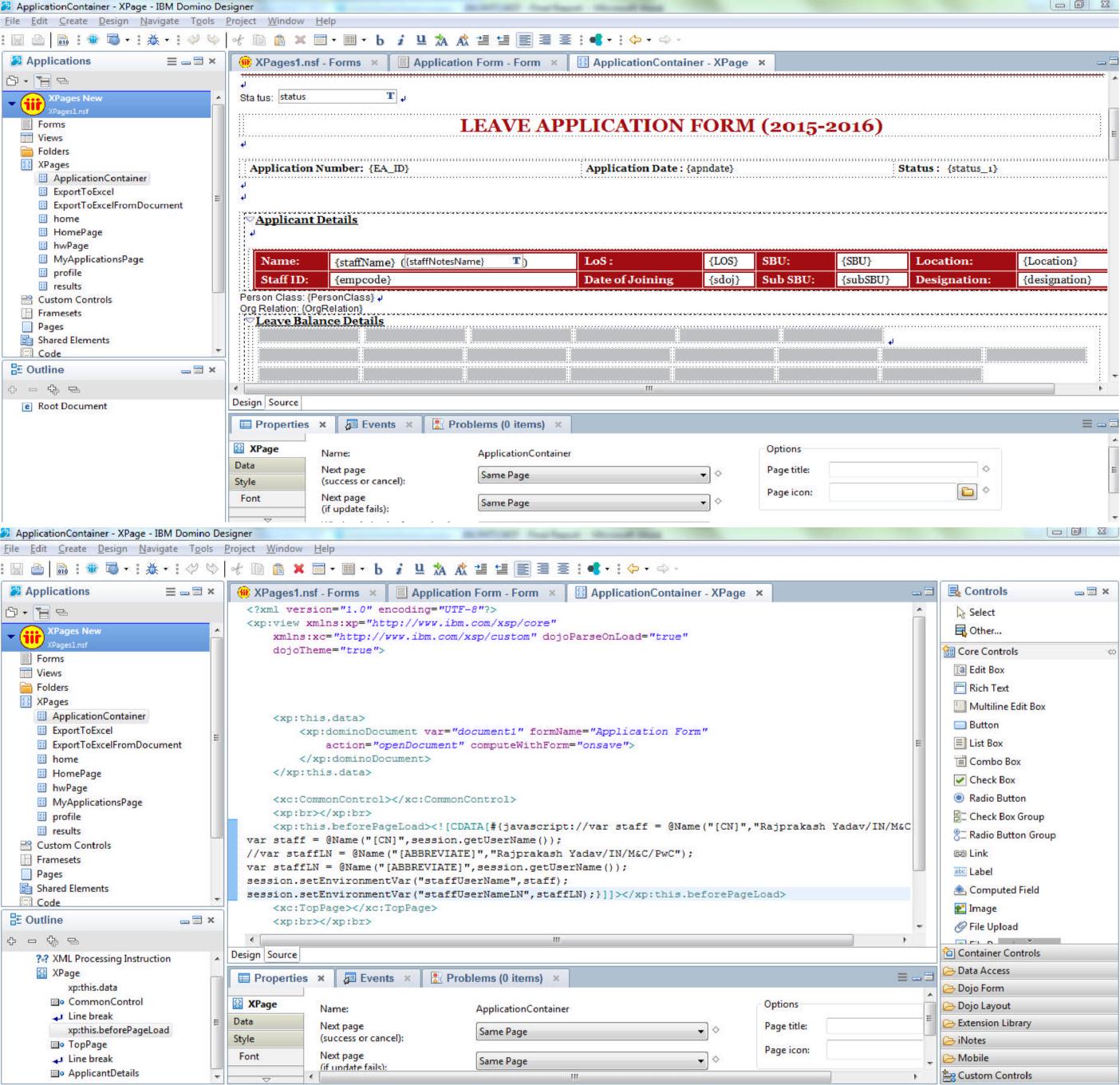


**Figure 17: Basic IBM Notes Form Design**

**Page 22 of 40**

**4.1.1 Why XPages?**

IBM Bluemix supports XPages and provides runtime environment for hosting and scaling the IBM Notes applications. Not only because Bulemix have XPage runtime have we decided to use XPage but it is compelling a new option for IBM Domino legacy applications. XPages is easy way to enhance [18] the IBM notes applications, making them faster, more reliable and better user experience. XPages allows us to add new functionality using best tools like JavaScript, Ajax, Dojo, and more. We can easily add Web 2.0 user experiences to our existing applications and we can take advantage of all of these powerful options either in the Notes client or on the Web.



**Figure 18: XPage Design and Source**

**Page 23 of 40**

**4.1.2 XPages Sample Code**

Sample code for application page:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<xp:view xmlns:xp=*"http://www.ibm.com/xsp/core"* xmlns:xe=*"http://www.ibm.com/xsp/coreex"*> <xp:this.data>

<xp:dominoDocument var=*"document1"* formName=*"Application Form"* action=*"openDocument"* computeWithForm=*"onsave"*>

</xp:dominoDocument> </xp:this.data>

<xp:panel>

<xp:section id=*"ApplicantDetails"* title=*"Applicant Details"* header=*"Applicant Details"* style=*"font-family:Georgia;font-size:10pt;font-weight:bold"*

initClosed=*"true"*>

<xp:br></xp:br> <xp:div align=*"left"*>

<xp:table style=*"width:100.0%;border-style:solid;border-*

*width:thin;border-color:rgb(161,13,18)"*

cellpadding=*"1"* cellspacing=*"1"* border=*"1"*> <xp:tr>

<xp:td

style=*"font-family:Georgia;font-size:10pt;background-color:rgb(161,13,18);color:rgb(255,255,255);border-color:rgb(161,13,18);border-style:solid;border-width:thin"*>

<xp:span style=*"font-weight:bold"*> Name:

</xp:span> </xp:td>

<xp:td

style=*"font-family:Georgia;font-size:10pt;border-color:rgb(161,13,18);border-style:solid;border-width:thin"*>

<xp:text escape=*"true"* id=*"staffName"*>

<xp:this.value><![CDATA[#{javascript:try{ if(document1.isNewNote()){ session.getEnvironmentString("staffUserName");

}else{

document1.getItemValue("staffName");

}

}catch(e){

}}]]></xp:this.value></xp:text> &#160;(<xp:inputText id=*"staffNotesName"* readonly=*"true"*>

<xp:this.value><![CDATA[#{javascript:try{ session.getEnvironmentString("staffUserNameLN");

**Page 24 of 40**

}catch(e){ }}]]></xp:this.value>

<xp:openPage name=*"/MyApplicationsPage.xsp"*></xp:openPage> </xp:actionGroup>

</xp:this.action> <xp:this.script>

<xp:scriptGroup>

-----------Script for all actions-------------------------->

</xp:this.script></xp:eventHandler></xp:button>&#160;&#160; <xp:button value=*"Edit"* id=*"Edit"* style=*"font-*

*family:Georgia"*><xp:this.rendered><![CDATA[#{javascript:!document1.isEditable() ||getComponent("status").value == "Approved" || getComponent("status").value == "Rejected"}]]></xp:this.rendered><xp:eventHandler event=*"onclick"* submit=*"true"* refreshMode=*"complete"*>

<xp:this.action>

<xp:changeDocumentMode mode=*"edit"*></xp:changeDocumentMode> </xp:this.action></xp:eventHandler></xp:button>

&#160;

<xp:button id=*"button2"* style=*"font-family:Georgia"*><xp:this.value><![CDATA[Save &Exit]]></xp:this.value><xp:this.rendered><![CDATA[#{javascript:document1.isEditable() && getComponent("status").value == "Initiate"}]]></xp:this.rendered><xp:eventHandler event=*"onclick"* submit=*"true"* refreshMode=*"complete"*>

<xp:this.action>

<xp:actionGroup>

<xp:openPage name=*"/MyApplicationsPage.xsp"*></xp:openPage> </xp:actionGroup>

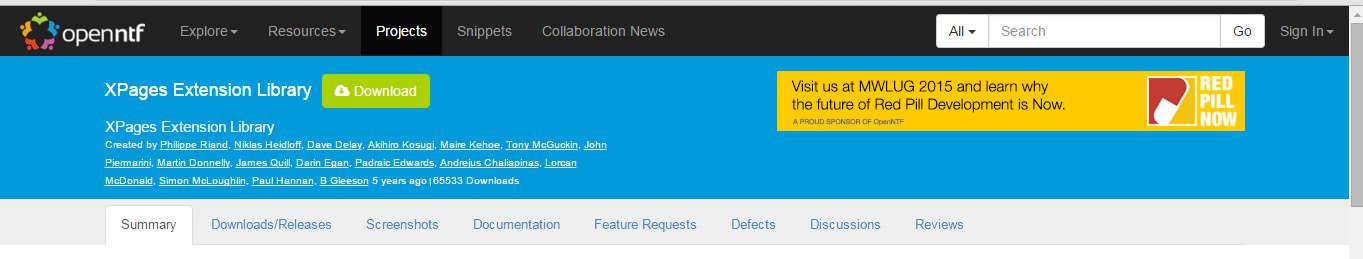
</xp:this.action></xp:eventHandler></xp:button> &#160;</xp:panel>

<xp:br></xp:br> <xp:br></xp:br> </xp:view>

**Page 25 of 40**

**4.2 Extension Library**

The XPages Extension Library provides ready to use controls. In order to build an XPages application, we need to install extension library on our local/server IBM Domino designer. The XPages Extension Library is provided as an open source project at OpenNTF [19] and includes the source code.



**Figure 19: openntf XPages Extension Library web page**

The XPages Extension Library must be enabled in each application. We can enable the extension library using below steps –

1. Open Application Properties for the application
2. Select the Advanced tab and
3. Select com.ibm.xsp.extlib.library under XPage Libraries

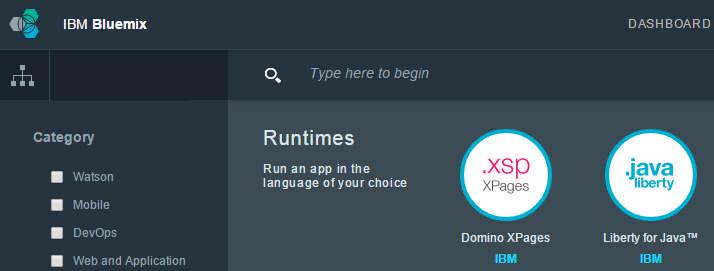
**Page 26 of 40**

**Chapter 5: Domino XPages in Bluemix**

With new XPages runtime, IBM Bluemix also introduced new Domino Data Service and new XPages boilerplates [17]. To maximize the benefit of cloud, XPages applications that going to be hosted on the IBM Bluemix are split into two parts – application code and data. The Bluemix architecture requires this particular separation.

**5.1 XPages Runtime**

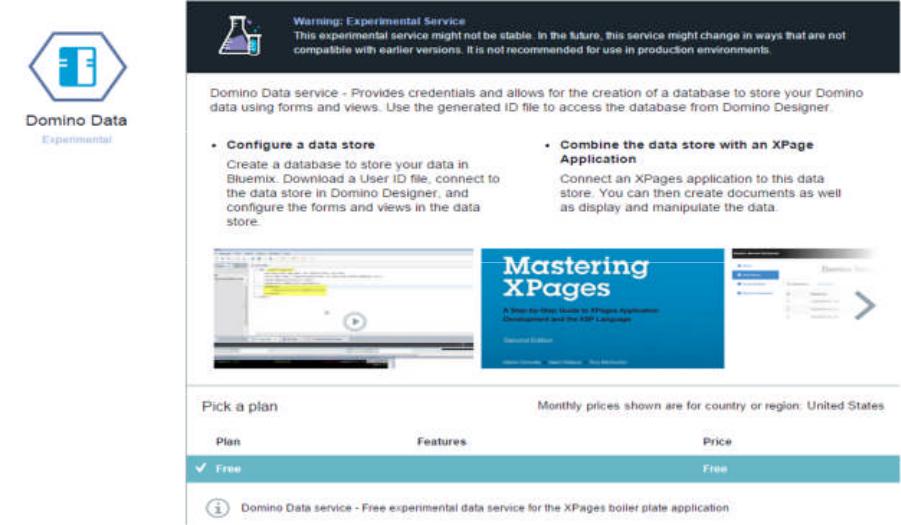
Xpages runtime runs the XPages application code on Bluemix. Code can be developed and tested on local domino servers. Once ready, this can be pushed to Bluemix using the cf command tool.



**Figure 20: XPages Runtime**

**5.2 Domino Data Service**

The Domino data service allows XPages applications running on Bluemix to store data in Domino NoSQL databases.

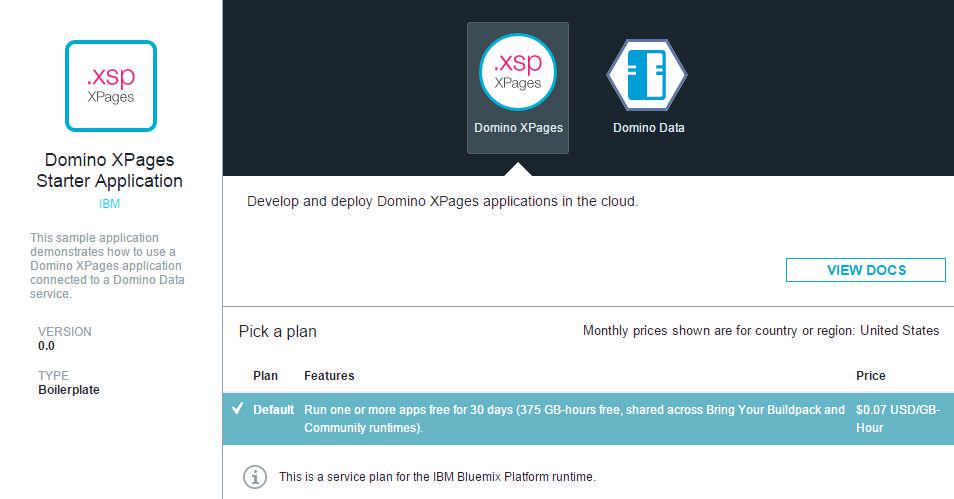


**Figure 21: Domino Data Service**

**Page 27 of 40**

**5.3 XPages Boilerplates**

In order for developers to get started easily an XPages boilerplate is provided. The boilerplate comes with the XPages runtime, the Domino data service and a sample. The sample is an easy to do application which shows CRUD operations, views and a responsive design.



**Figure 22: XPages Boilerplates**

**Page 28 of 40**

**Chapter 6: Testing and Integration**

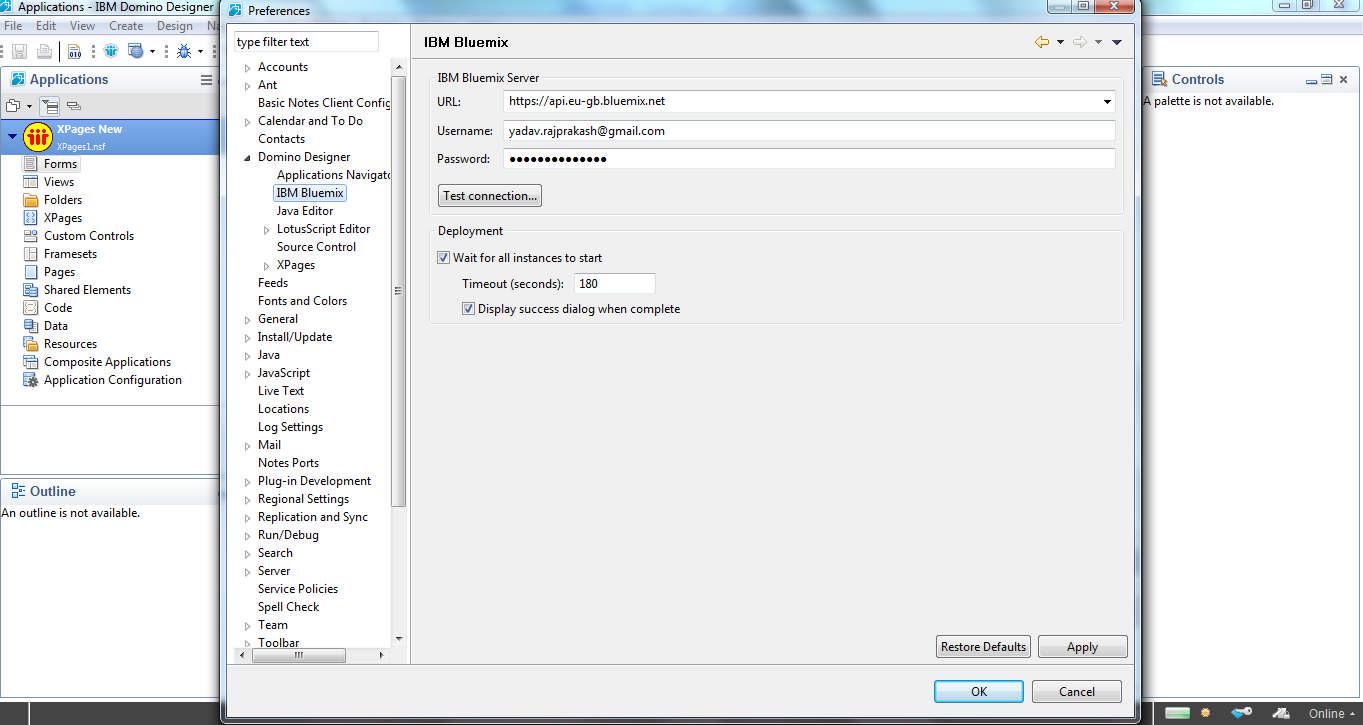
As we discussed in last chapter, there are different options for Domino XPages in Bluemix. Based on previous understanding and options available in IBM Bluemix, we can use the XPages for Domino on Bluemix to [17] –

1. **Enhance our existing applications** – We can create cool responsive versions ofIBM Notes client applications for cloud.
2. **To always have latest XPages runtime** for our applications. Bluemix will havelatest libraries.
3. **To Scale applications** – PaaS feature of Bluemix provides capability of scaling appsand recovering without loss.

**6.1 Creating New Applications in Bluemix**

In order to make use of integrated Bluemix tool [20] for IBM Designer, we must have installed latest XPages Extension library. For creating applications in Bluemix, we must have IBM Bluemix account created. There is one month trial option available under which we use certain services for free. Once we are logged in, follow below steps to use the XPages runtime from the Bluemix dashboard –

1. Click on Catalog and then scroll to the end. Click Bluemix Labs Catalog
2. Select the XPages Web Starter from the Boilerplates section of the Bluemix Labs Catalog, and click View More.
3. Provide the application name, and click Create.
4. From the Bluemix dashboard, click on the created application
5. From the left navigation bar, click on Start Coding and then select CF Command Line Interface.
6. Click Download Starter Code and save the code to local directory.
7. Add Bluemix credentials to Designer to use Designer's built-in Bluemix tools

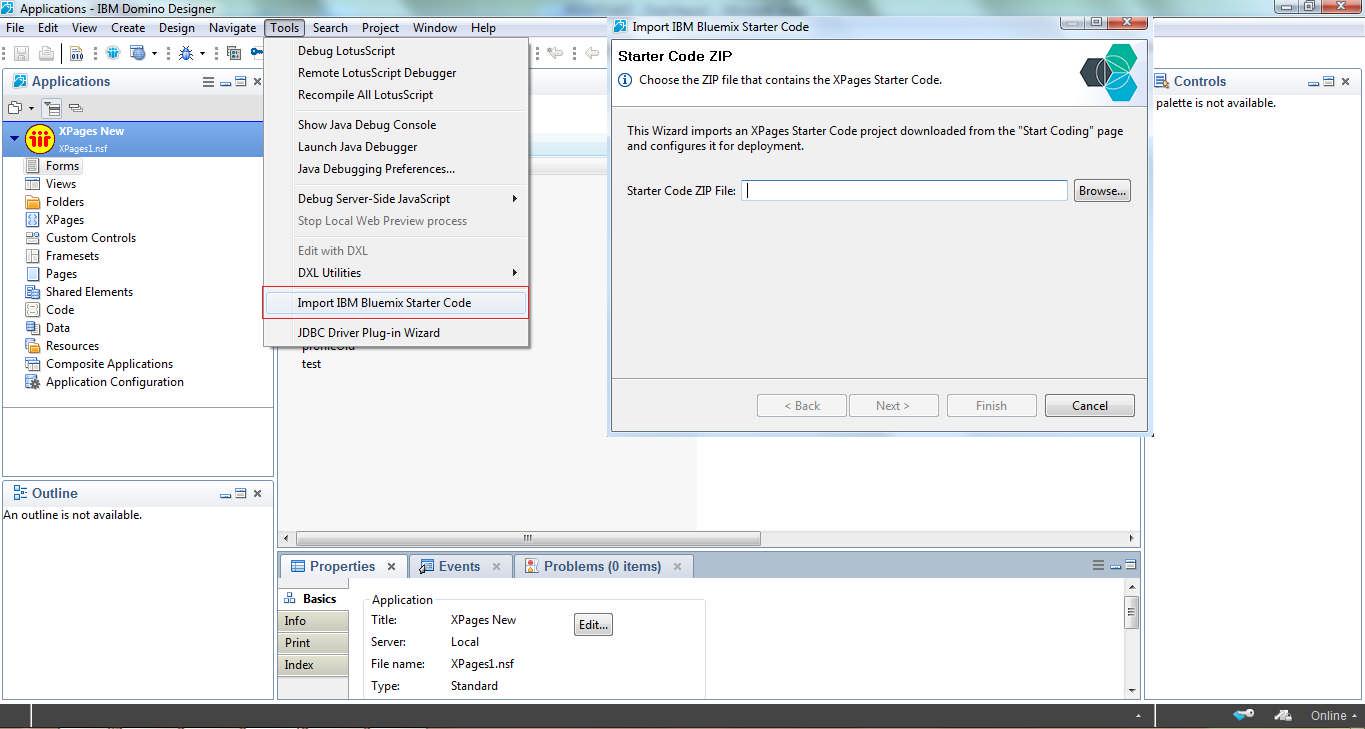


**Figure 23: Built-In Bluemix to Designer**

**Page 29 of 40**

* 1. Click File > Preferences. The Preferences window appears
  2. Click Domino Designer > IBM Bluemix
  3. In the Bluemix Server section, enter the URL of the Bluemix server and Bluemix credentials.
  4. Click Test connection to verify that Designer can connect to the Bluemix server.
  5. Click Apply and then click OK.

1. Import the started code to Bluemix. Click on Tools -> Import IBM Bluemix Starter Code



**Figure 24: Import IBM Bluemix Starter Code**

1. Modify the application as needed and redeploy the application to Bluemix using the integrated Bluemix Tools.

**6.1.1 IBM Bluemix Manifest Code**

applications:

* path: . memory: 256M instances: 1

domain: eu-gb.mybluemix.net name: LMSStage

host: LMSStage disk\_quota: 1024M services:

**Page 30 of 40**

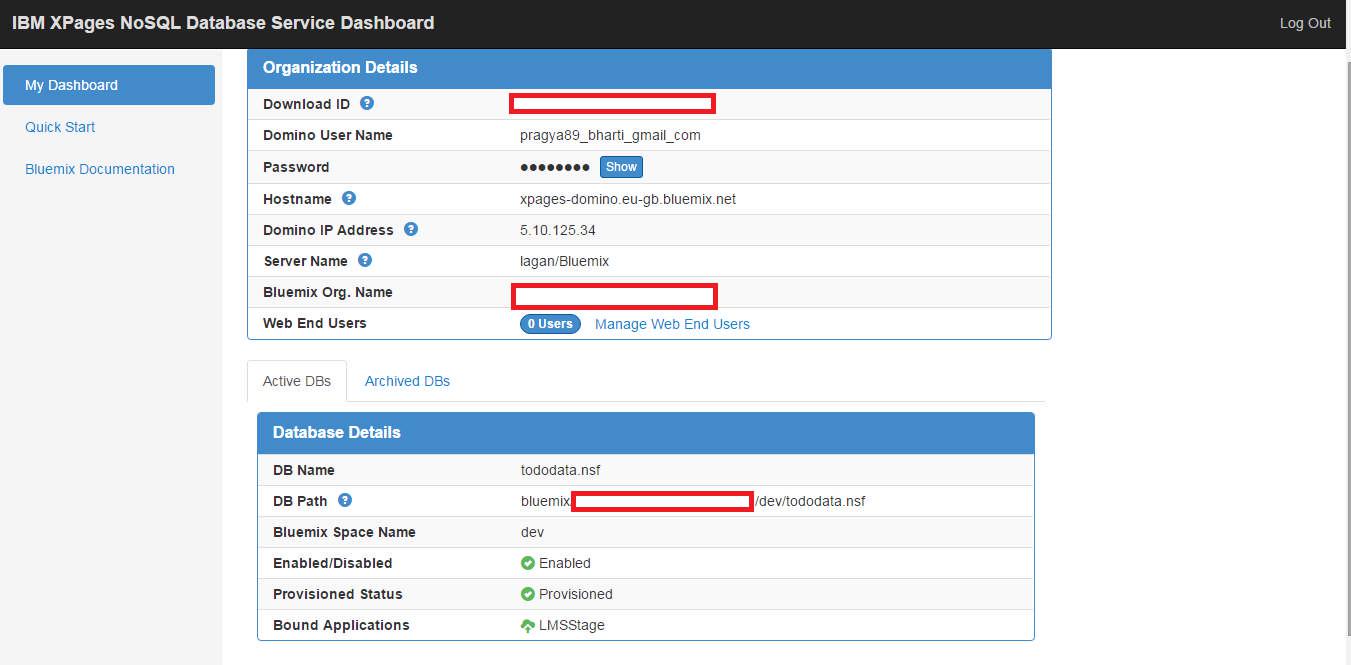
* + LMSStage-XPagesData env:

APP\_HOME\_URL: /ToDo\_design.nsf APP\_PRELOAD\_DB: ToDo\_design.nsf buildpack: xpages\_buildpack

1. **Pushing Existing Applications to Bluemix**

Legacy IBM Notes applications generally contain both the application design and data. As we discussed in previous chapters, we need to separate out the design and data in order to make use of benefit of cloud in Bluemix. We have followed the below steps to split the local copy of existing applications before deploying to Bluemix.

1. Convert the legacy application into corresponding XPages application.
2. Create a XPages NoSQL Database service instance
   1. In the Bluemix Dashboard, create an instance of the XPages NoSQL Databaseservice.
   2. In the Services section of the Dashboard navigation bar, click on the instance of the XPages NoSQL Database service.
   3. Click Launch.
   4. The XPages NoSQL Database Service Dashboard opens in a new browser window and displays our credentials.



**Figure 25: NoSQL Database Service Dashboard**

1. Click the ID Download link to save the credentials locally and record the server and database name and location.

**Page 31 of 40**

1. Split the application into design and data.
   1. Open the application in the designer.
   2. Right-click on the application in the Applications Navigator and select Application > New Copy to create a duplicate of the application. Name one copy MLS\_design.nsf and the other copy LMS\_data.nsf. Open both applications in the designer.
   3. Add the user name generated by the XPages NoSQL Database Service Dashboard to the Access Control List and provide Designer access.
   4. Switch to the service-provided Notes ID by clicking File > Security > Switch ID.
   5. From the data application (LMS\_Data.nsf), delete all XPages and Custom Controls.
   6. In the design application (LMS\_Design.nsf), delete all Forms and Views.
   7. Sign both the applications with the ID provided by Bluemix service.
2. Replace the data references with Bluemix context. a. <xp:DominoView> as:

<xp:dominoView databaseName=*"#{javascript:bluemixContext.isRunningOnBluemix()?*

*bluemixContext.getDataService().findDatabaseName() : 'LMS\_Data.nsf'}"*>

1. @DbName as:

var one = @Unique(@DbColumn(bluemixContext.isRunningOnBluemix()? bluemixContext.getDataService().atDbName():@DbName,"vwEmployeeData", 1));

1. Change all the reference to Data copy:

if( null == requestScope.dataServiceDatabase) { if (bluemixContext.isRunningOnBluemix()) {

var otherDatabaseVector = bluemixContext.getDataService().atDbName(); var otherDatabaseServer = otherDatabaseVector.get(0);

var otherDatabaseAppPath = otherDatabaseVector.get(1); var otherDatabase =

session.getDatabase(otherDatabaseServer,otherDatabaseAppPath); requestScope.dataServiceDatabase = otherDatabase;

} else {

requestScope.dataServiceDatabase = session.getDatabase(null, "LMS\_Data.nsf");

}

}

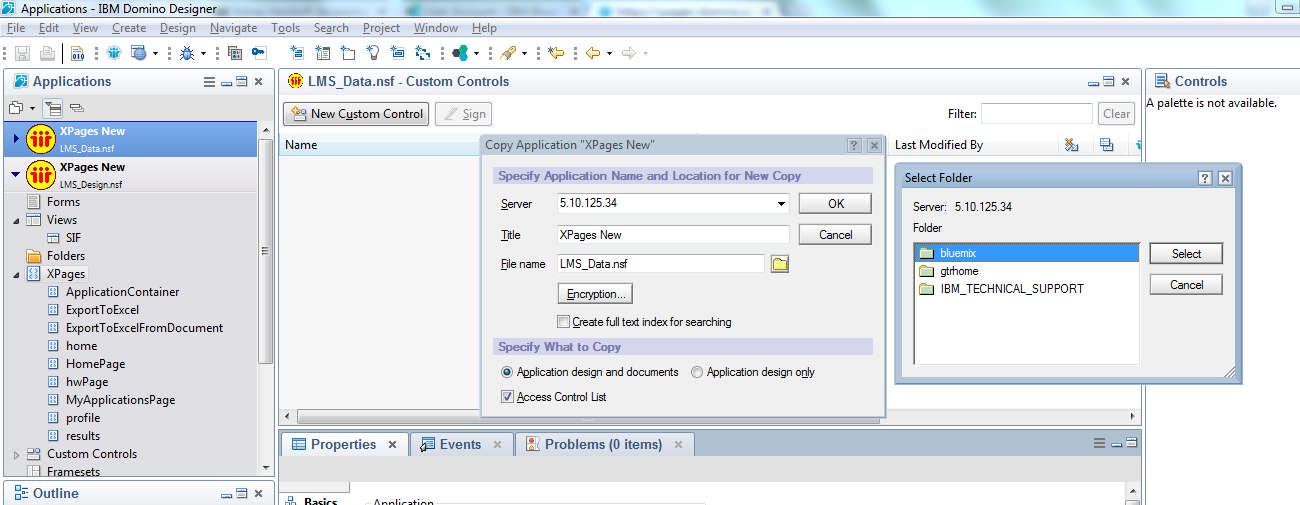
requestScope.dataServiceDatabase.getViews();

1. Deploy the data application to the service-provided Domino server

a. Right-click on the data application and choose Application > New Copy.

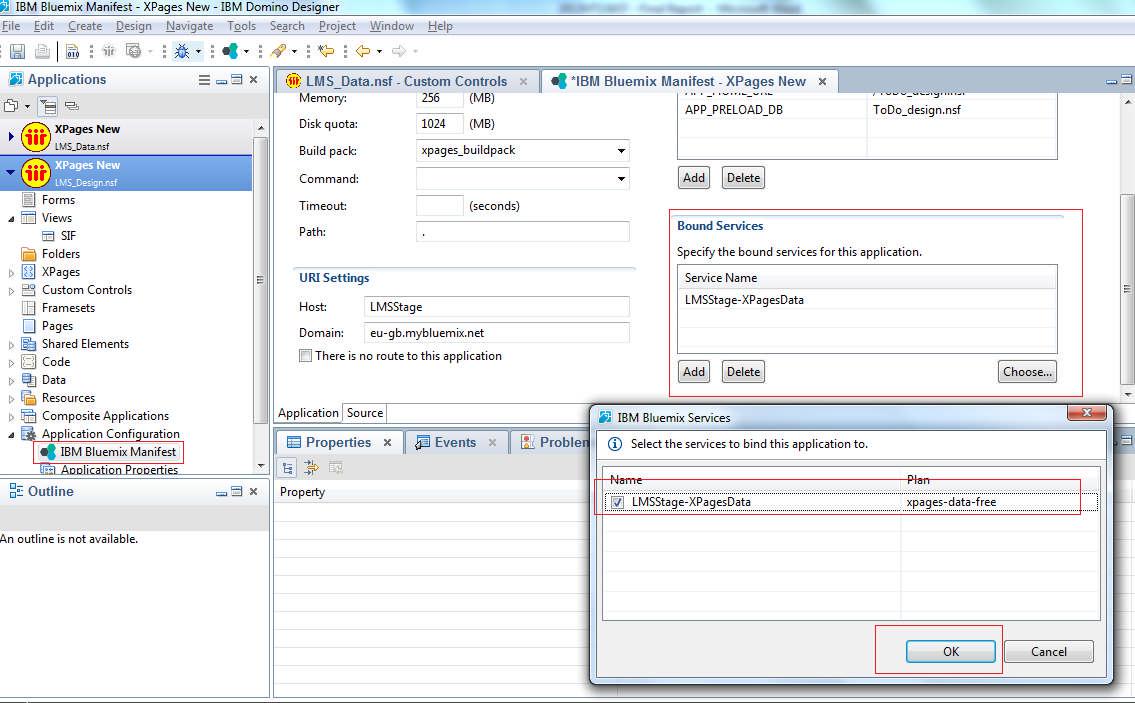
**Page 32 of 40**

1. In the Server and File field, enter the Domino Server name and File name that was generated by the XPages NoSQL Database Service Dashboard.



**Figure 26: NoSQL Data Deployment**

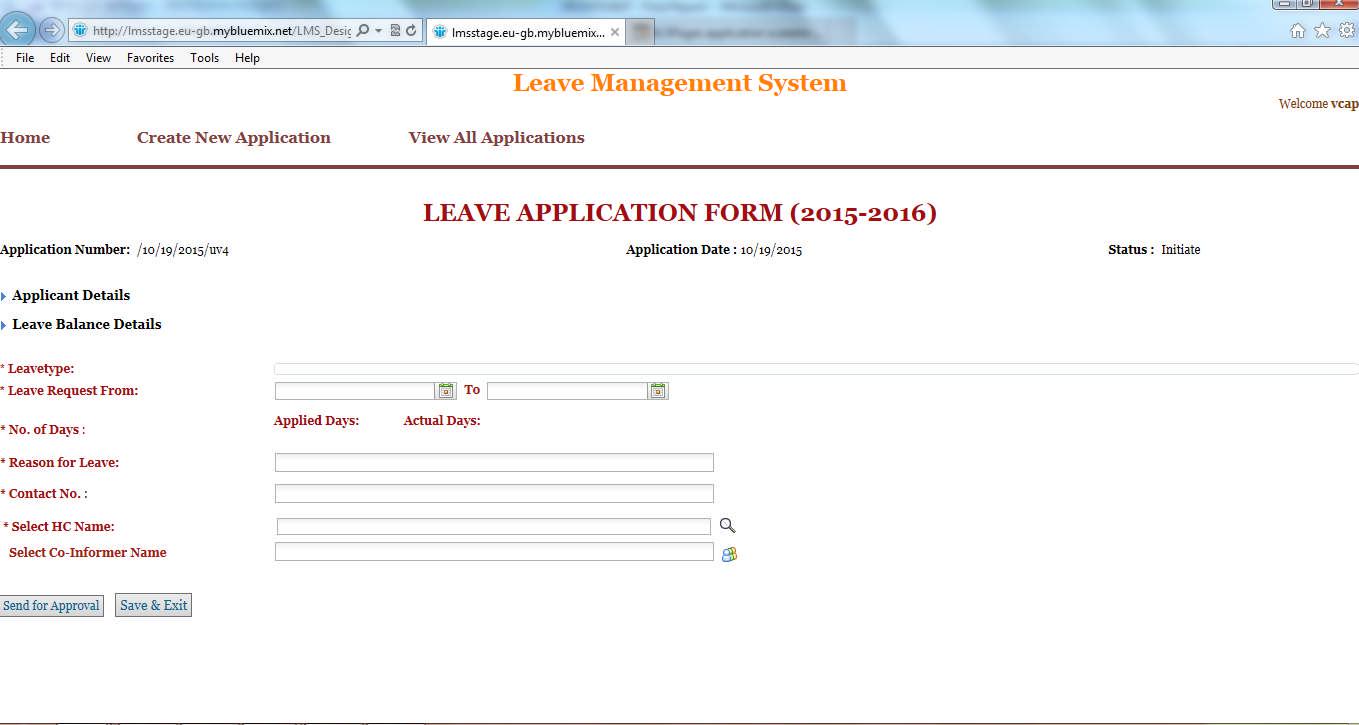
1. Deploy the design application to the Bluemix
   1. Open the design application in Domino Designer and select the application and click the Deploy to IBM Bluemix button in the toolbar.
   2. Open the IBM Bluemix Manifest file and set the Bound Services.



**Figure 27: Bound the Services to IBM Bluemix**

**Page 33 of 40**

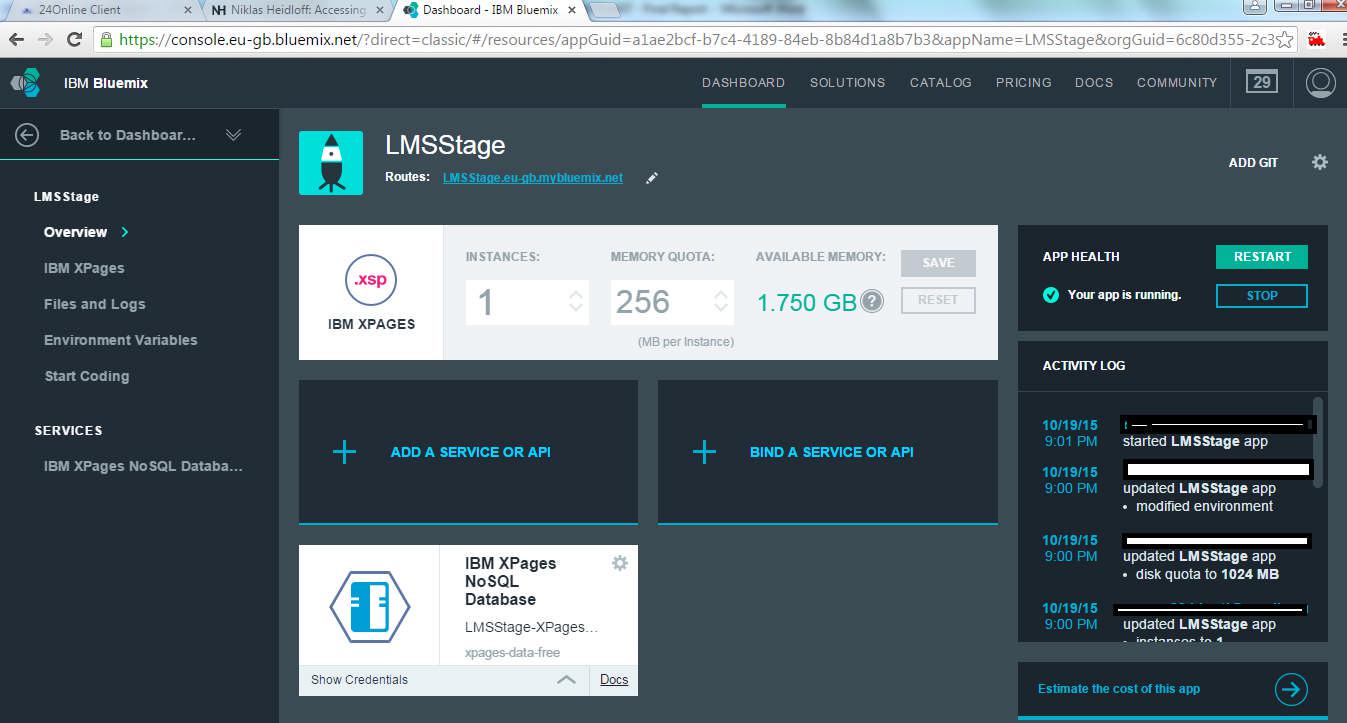
1. Open the application in browser using the link provided - http://lmsstage.eu-gb.mybluemix.net/LMS\_Design.nsf/ApplicationContainer.xsp



**Figure 28: Sample Application on IBM Bluemix**

**6.3 Scaling XPages Application in IBM Bluemix**

Applications deployed to Bluemix are scalable just like any other runtime in Bluemix. On IBM Bluemix, XPages applications are deployed in separate layers i.e. data and design. We can bind the Domino Data Service to XPages applications. In the IBM Bluemix Dashboard, open the application and add service or API.



**Figure 29: Scaling the XPages Application**

**Page 34 of 40**

**Summary**

IBM Notes is one of the oldest collaboration technology and many big organizations have in their nerves. We have seen an organization having more than 80K IBM Notes applications and every day counting. Not only IBM notes applications are involved in daily work but also it’s really easy to build a workflow based application. With recent buzz on cloud and many successful transitions of legacy applications to cloud, business organizations are looking for more opportunities on cloud. Businesses that have legacy applications as IBM Notes, now want to move to cloud and looking to make use of this opportunity.

As discussed in previous chapters, moving the IBM Notes applications to cloud is not easy transition. Not all cloud service provider support the IBM Notes Domino infrastructure. Only way is to migrate the legacy application to any other platform like java, .Net, SharePoint, etc. These legacy applications and capabilities such as built-in workflow, role-based security and business logic must be redeveloped.

Redeveloping a Lotus Notes application requires redesigning the whole application design and then the big task of transferring the data. Instead of migrating the applications to another platform, we should consider the XPages applications. IBM Bluemix introduced XPages runtime, Domino Data Service and XPgaes boilerplates. This is clearly best option for IBM Notes legacy applications if an organization wants to move to cloud. We have used the XPages runtime for creating new application on cloud and also make use of Domino Data Services for migrating the existing applications to cloud. We have separated out the data and design of the application and this separation provides the benefits of cloud on IBM Bluemix.

**Page 35 of 40**

**Conclusion and Recommendations**

In this work our main focus was to prepare a proof of concept for making the domino applications cloud ready. With the previous chapters, clearly moving the legacy IBM Notes applications to cloud is very tedious and challenging work. Many authors already have presented white papers on capabilities of cloud platform to host these domino applications. Considering all these results of the white papers and study we have done in this work, XPages is best approach for moving the legacy IBM Notes applications to cloud.

IBM Domino and Notes business users should consider enabling their application to XPages. This comes with lots of benefit like it will reduce the need of data migration, reducing the time needed to redeveloping the Notes features, etc.



**Figure 30: Migration Comparison based on different platforms**

Above comparison clearly shows that XPages enablement is better approach. This approach surely reduces the risks involved in migrating the IBM notes applications to cloud. Data migration is no more part of the cloud enablement. This way we can save lot of efforts of redevelopment of Notes features and UI modifications.

With “Cloud” becoming an increasingly viable delivery mechanism for providing robust services around mail, calendaring and contacts, organizations of all sizes can now move some or all applications to the cloud at a comfortable rate and pace.

**Page 36 of 40**

**Directions for future work**

The XPages approach still poses the risk of business disruption and requires significant cost and efforts. Currently, IBM Bluemix is only platform which provides the XPages runtime for domino applications and that is also in experimental lab section. This means that IBM Bluemix still not supporting the XPages and other related technologies in regular support and we cannot deploy the applications to production. All results here are results of staging deployment of sample applications.

This work is completely done on IBM Bluemix and in future this work can be carried out at other PaaS systems too. Microsoft Azure, Google App Engine and many more cloud service providers have great PaaS tools and in future we can try to enable these XPages application on those platforms.

XPages enablement is also very exhaustive process as it requires a lot of efforts in modifying the UI and business logic. We can develop a XPages automatic converter for IBM Notes applications. This will surely further reduce the risks and business disruptions. Manual XPages enablement also requires staff and other resources. Automating this process will reduce the staff requirement.

In this work we just have touched the application development part; there are many other aspects like Domino Server, Domino Administration, etc. In order to integrate the on premise applications with cloud hosted apps, we need to make sure that these all aspects are analyzed properly.

IBM Bluemix provided many other services; we can make use of all the PaaS services in our applications and enhance the business applications. We can make use of all cloud benefits on IBM Bluemix in our applications. With growing mobile and handheld business users, IBM Notes applications too should be enabled for these features. IBM Bluemix provides mobile computing services and same can be integrated in our applications.

**Page 37 of 40**

**References**

1. IBM Notes and Domino Wiki, http://www-10.lotus.com/ldd/dominowiki.nsf/
2. What is Cloud Computing? , http://in.pcmag.com/networking-communications-software/38970/feature/what-is-cloud-computing
3. Cloud Computing Wikipedia, https://en.wikipedia.org/wiki/Cloud\_computing
4. NIST Cloud Computing Program, http://www.nist.gov/itl/cloud/
5. Rajkumar Buyya, Andrzej M. Goscinski and James Broberg. Cloud Computing : Principles and Paradigms. Wiley, December, 2010
6. Microsoft Azure Wikipedia, https://en.wikipedia.org/wiki/Microsoft\_Azure
7. Azure Storage Documentation, https://azure.microsoft.com/en-in/documentation/services/storage/
8. Google App Engine: Platform as a Service, https://cloud.google.com/appengine/docs
9. IBM Bluemix Docs, https://www.ng.bluemix.net/docs/
10. Cloud Foundry, https://www.cloudfoundry.org/
11. Cloud Native Applications on Bluemix - http://www.redbooks.ibm.com/redpieces/abstracts/sg248275.html?Open
12. Building Cloud Native Applications, Ryan Baxter, http://ryanjbaxter.com/2015/07/13/building-cloud-native-applications/
13. IBM, Microservices from Theory to Practice, IBM Redbooks, August, 2015
14. Heroku, Cloud Application Platform, https://www.heroku.com/
15. The Twelve-Factor App, http://12factor.net/
16. Migrating IBM Lotus Notes and Domino to the Cloud: Selecting Microsoft versus Google, December, 2010
17. IBM ConnectED 2015, Domino XPages for IBM Bluemix, http://www.ibm.com/developerworks/cloud/library/cl-bluemixfoundry/
18. Responsive Design, http://whatis.techtarget.com/definition/responsive-design
19. OpenNTF, http://www.openntf.org/
20. IBM Bluemix, The Digital Innovation Platform, https://www.ng.bluemix.net
21. 5 Things to Know about Why You Should Register for Bluemix, https://www.ibm.com/developerworks/community/blogs/5things/entry/5\_things\_to\_know\_ about\_why\_you\_should\_register\_for\_bluemix?lang=en

**Page 38 of 40**

**Checklist for the items in the report**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. | **Is the final report neatly formatted with all the elements required** | | Yes |  |
|  | **for a technical Report?** | |  |  |
|  |  | |  |  |
| 2. | Is the Cover page in proper format as given in Annexure A? | | Yes |  |
|  |  | |  |  |
| 3. | Is the Title page (Inner cover page) in proper format? | | Yes |  |
|  |  |  |  |  |
| 4. | (a) | Is the Certificate from the Supervisor in proper format? | Yes |  |
|  | (b) | Has it been signed by the Supervisor? | Yes |  |
|  |  | |  |  |
| 5. | Is the Abstract included in the report properly written within one page? | | Yes |  |
|  | Have the technical keywords been specified properly? | | Yes |  |
|  |  | |  |  |
| 6. | Is the title of your report appropriate? **The title should be adequately** | | Yes |  |
|  | **descriptive, precise and must reflect scope of the actual work done.** | |  |  |
|  | Uncommon abbreviations / Acronyms should not be used in the title | |  |  |
|  |  | |  |  |
| 7. | Have you included the List of abbreviations / Acronyms? | | Yes |  |
|  |  | |  |  |
| 8. | Does the Report contain a summary of the literature survey? | | Yes |  |
|  |  | |  |  |
| 9. | Does the Table of Contents include page numbers? | | Yes |  |
|  | (i). | Are the Pages numbered properly? (Ch. 1 should start on Page # 1) | Yes |  |
|  | (ii). | Are the Figures numbered properly? (Figure Numbers and Figure | Yes |  |
|  |  | Titles should be at the bottom of the figures) |  |  |
|  | (iii). Are the Tables numbered properly? (Table Numbers and Table Titles | | Yes |  |
|  |  | should be at the top of the tables) |  |
|  | (iv). | Are the Captions for the Figures and Tables proper? | Yes |  |
|  | (v). | Are the Appendices numbered properly? Are their titles appropriate | Yes |  |
|  |  | |  |  |
| 10. | Is the conclusion of the Report based on discussion of the work? | | Yes |  |
|  |  | |  |  |
| 11. | Are References or Bibliography given at the end of the Report? | | Yes |  |
|  | Have the References been cited properly inside the text of the Report? | | Yes |  |
|  | Are all the references cited in the body of the report | | Yes |  |
|  |  | |  |  |
| 12. | Is the report format and content according to the guidelines? The report | | Yes |  |
|  | should not be a mere printout of a Power Point Presentation, or a user | |  |  |
|  | manual. Source code of software need not be included in the report. | |  |  |
|  |  |  |  |  |

**Page 39 of 40**

