Microsoft

Packaging and Deploying 5 Java Frameworks in 5 Ways into Azure

Ed Burns

@edburns

Ed Burns Credentials

Client

NCSA Mosaic (1994)

SGI Cosmo Web Authoring (1995)

Sun Netscape 6 OJI Applet Engine (1999)

Server

J2EE JSF Spec Lead (2002 - 2019)

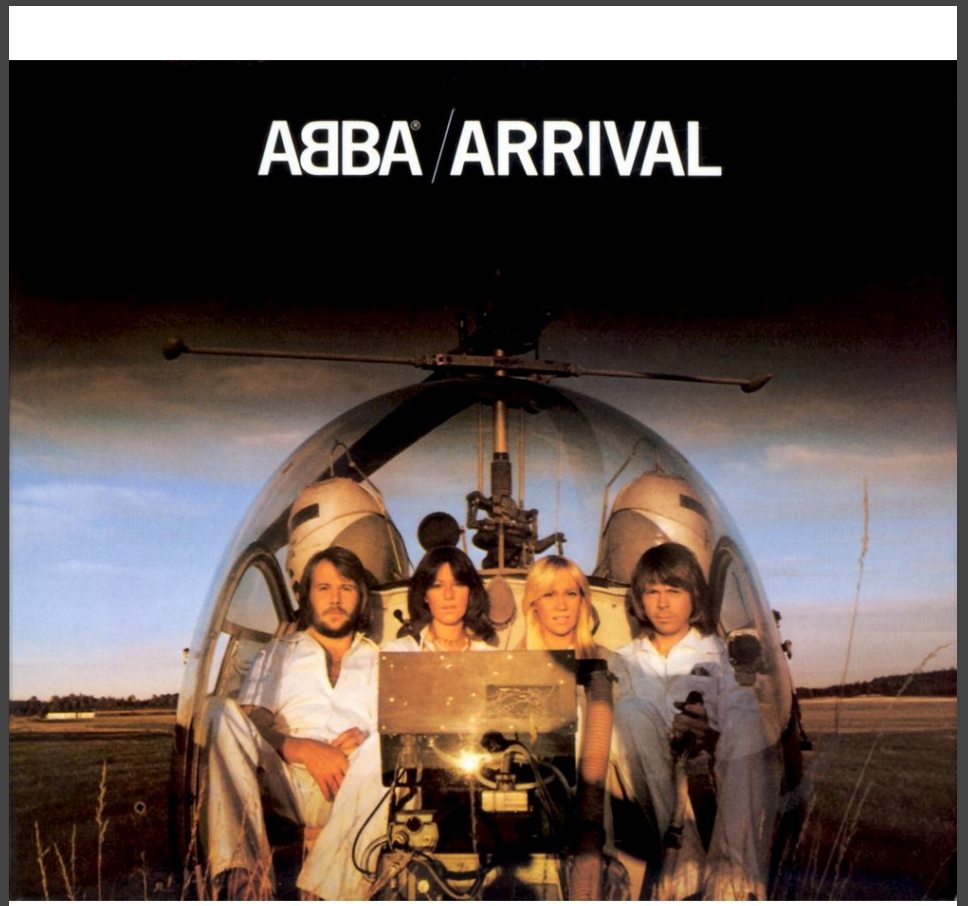
Oracle Java EE

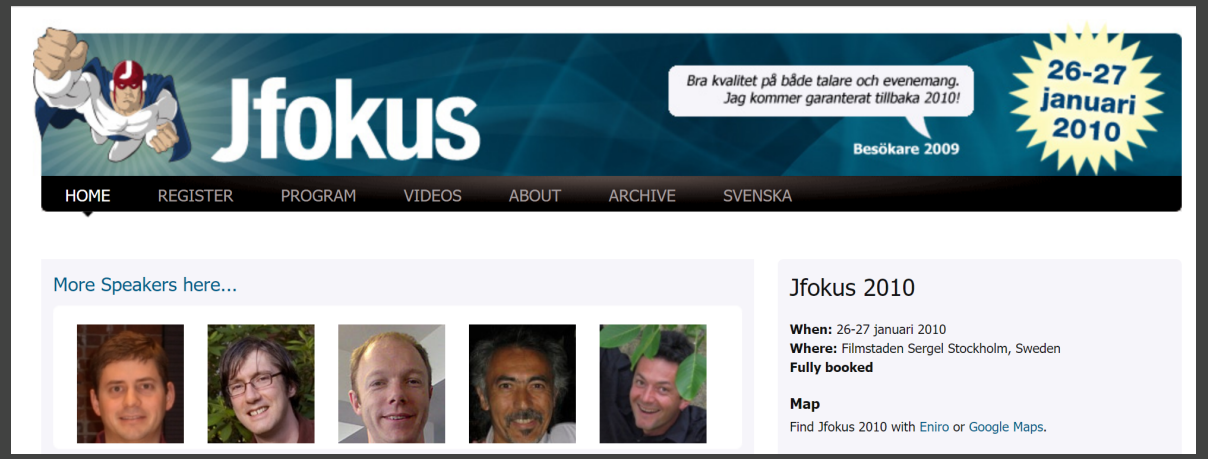
Servlet Spec Lead, JSF, Bean Validation, etc.

Microsoft Azure Cloud (2019)

Books

My First Time in Sweden





Kaviar?

My Plan for Your Time Investment

- Brief Personal Journey
- Java On Azure
- The Five Frameworks
- The Five Ways to Azure

Microsoft

Azure

Brief Personal Journey

- You must constantly adapt
- Not necessarily frequently change jobs
- Companies need to adapt too

Starting on the Client: Birth of a Big Thing

First Graphical Web Browser

Gratis and Free software

Built on prior work really well

Ubiquitous (cross platform)

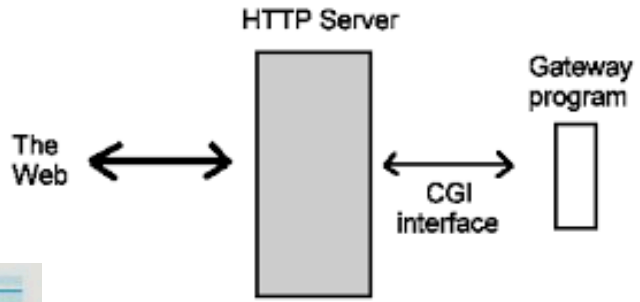
Delivered something everyone wanted

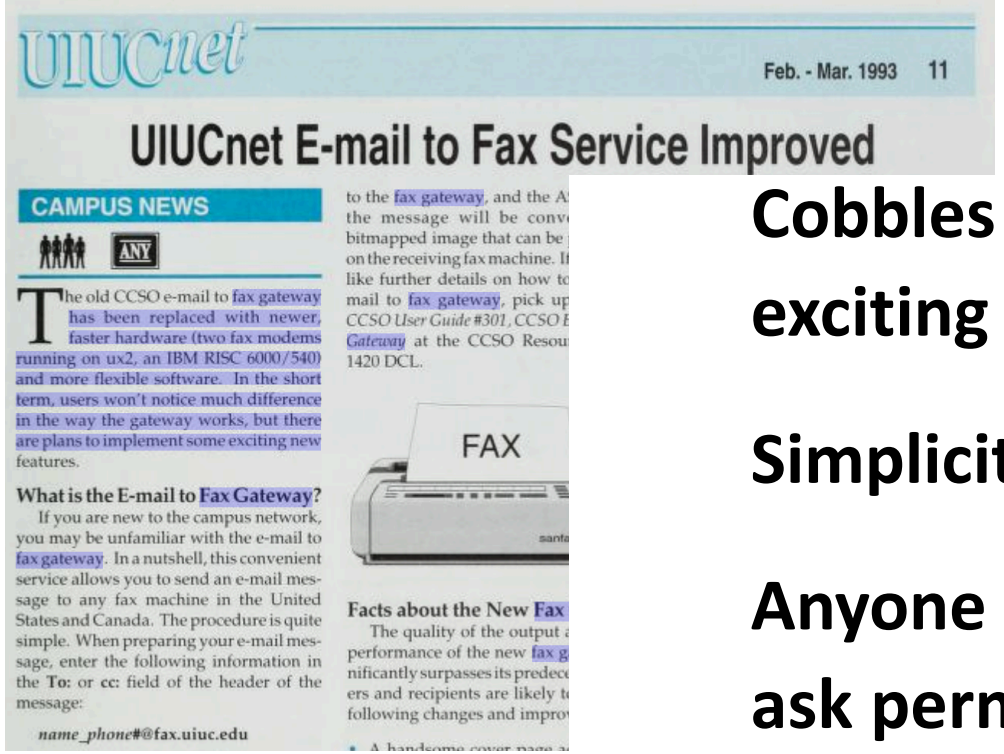
Easy to author: copy and paste from view source!

Oil and Chemistry Building at University of Illinois in Urbana-Champaign

Home of NCSA Mosaic

Dabbling on the Server





UIUCnet Feb. - Mar. 1993 11

UIUCnet E-mail to Fax Service Improved

CAMPUS NEWS

The old CCSO e-mail to fax gateway has been replaced with newer, faster hardware (two fax modems running on ux2, an IBM RISC 6000/540) and more flexible software. In the short term, users won't notice much difference in the way the gateway works, but there are plans to implement some exciting new features.

What is the E-mail to Fax Gateway?

If you are new to the campus network, you may be unfamiliar with the e-mail to fax gateway. In a nutshell, this convenient service allows you to send an e-mail message to any fax machine in the United States and Canada. The procedure is quite simple. When preparing your e-mail message, enter the following information in the **To:** or **cc:** field of the header of the message:

name_phone#@fax.uiuc.edu

to the fax gateway, and the A
the message will be conv
bitmapped image that can be
on the receiving fax machine. I
like further details on how t
mail to fax gateway, pick up
CCSO User Guide #301, CCSO E
Gateway at the CCSO Resou
1420 DCL.

FAX

Facts about the New Fax

The quality of the output
performance of the new fax g
nificantly surpasses its predec
ers and recipients are likely t
following changes and impro

Cobbles existing things together in an exciting new way.

Simplicity: HTTP/1.0, TCP/IP sockets

Anyone can add new servers, no need to ask permission.

Leverages network effect

Birth of the Monolith: Spring and J2EE

Photo: Les Chatfield

Addressed shared pain points

Transparent development process

Spring rode the crest of vendor marketing efforts, differentiated with operational excellence

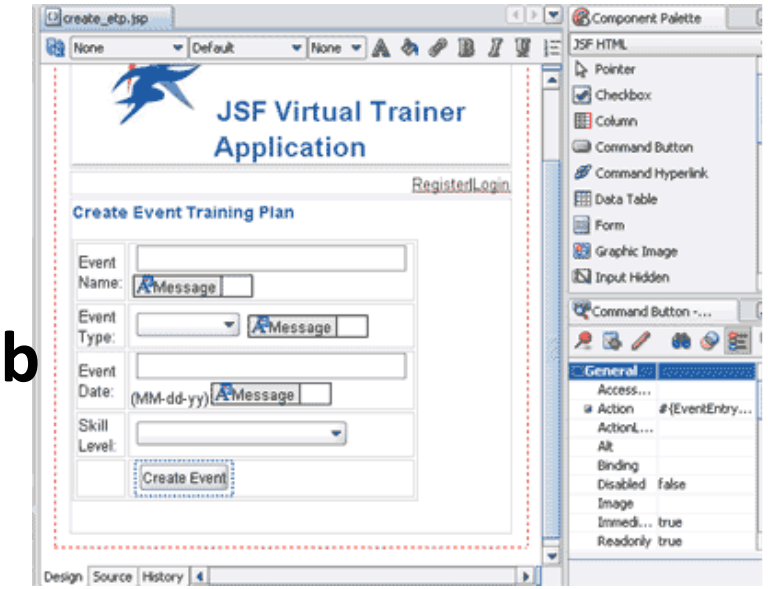
J2EE had multi-vendor concept, strong community governance

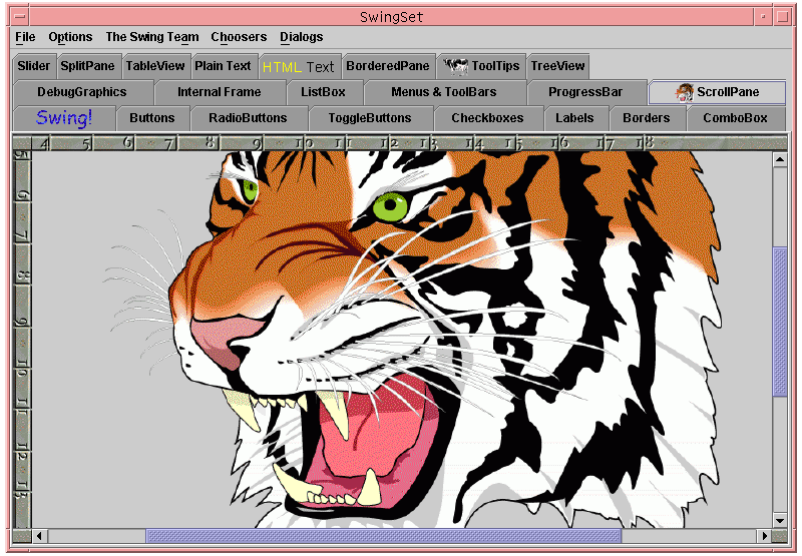
My First “War”: Web Frameworks

Good variety of “good enough” tools

Created a component ecosystem

Lots of buzz around this space due to “Web Framework Wars”





The Cloud: Birth of Another Big Thing

2004: Sun Utility Computing

before it’s time…and at the wrong home

I’m Ready for the Cloud

My response to cloud disruption

Get out from inside of the monolith

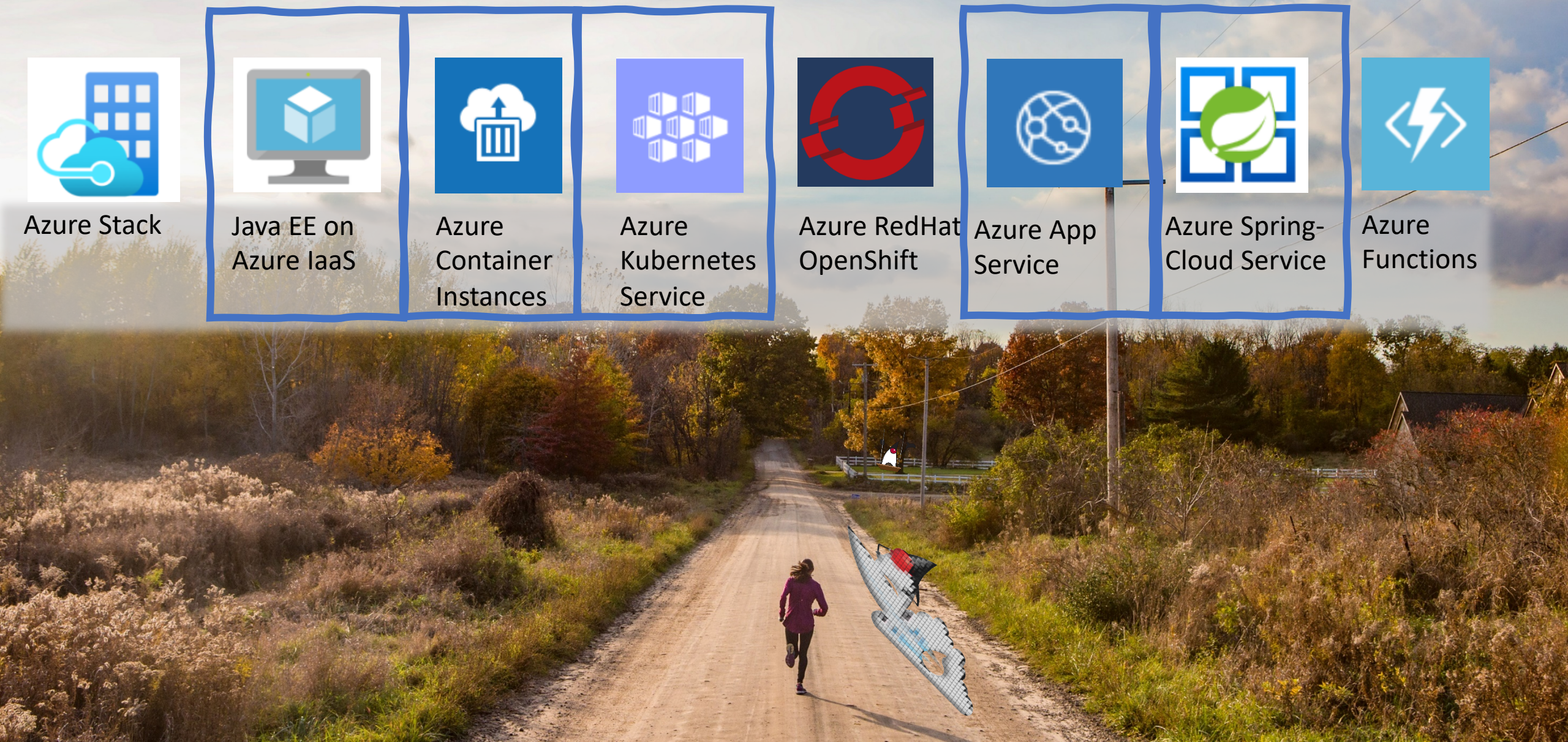
Work with the hosting platform

- Monolith
- Microservices

Helps enterprises scale

Java on Azure

Java on Azure for Wherever You Are in your Cloud Journey



Java Cloud Native on Azure – Landscape

Azure Services

CosmosDB | Service Fabric | App Service | Container Instances | Container Registry | Kubernetes Service | Serverless Functions | App Insights | VSTS DevOps | SQL Service | Event Grid | Event Hub | Service Bus | Active Directory

Multi-Cloud Platforms

OPENSHIFT by Red Hat
CLOUDFOUNDRY
Pivotal

JVMs, Frameworks, Runtimes

AZUL SYSTEMS
OpenJDK
ORACLE

Tools

JETBRAINS
Maven

DevOps

GitHub
splunk>
cloudbees
New Relic.

IaaS – Linux Distributions

RED HAT ENTERPRISE LINUX
ubuntu
CoreOS
ORACLE Linux
SUSE
debian

On Azure, over 50% of workloads are Linux-based.

Cloud Automation

HashiCorp Terraform
ANSIBLE

Kubernetes & Containers

kubernetes
docker

OSS Foundations

eclipse FOUNDATION MEMBER
THE APACHE SOFTWARE FOUNDATION FOUNDATION MEMBER

Microsoft SDKs for Java
Over 50+ services covered

Contemporary Java Frameworks

Contemporary Java Frameworks

- Examine some common aspects of each
- API Governance model
- Primary way people work with it
  - Writing code
  - Build system commonly used
- Basic idea of creating and running an app

Jakarta EE

- API Standard
  - JCP->Eclipse Specification Process
- IDE-based
- Build system
  - Whatever you like, mostly maven
- Create app
  - Run a maven archetype
- Run app
  - $ Deploy with a maven plugin

Java EE Simple App

```java
31  @Stateless
32  @Path("coffees")
33  public class CafeResource {
34
35      @GET
36      @Produces({ MediaType.APPLICATION_XML })      Reza Rahman, a month ago • Downgrading applic
37      public List<Coffee> getAllCoffees() {
38          return this.cafeRepository.getAllCoffees();
39      }
40
41      @POST
42      @Consumes({ MediaType.APPLICATION_XML })
43      public Response createCoffee(Coffee coffee, @Context UriInfo uriInfo) {
44          try {
45              coffee = this.cafeRepository.persistCoffee(coffee);
46              String path = uriInfo.getAbsolutePath() + "/" + coffee.getId();
47              return Response.created(URI.create(path)).build();
48          } catch (PersistenceException e) {
49              logger.log(Level.SEVERE, "Error creating coffee {0}: {1}.", new Object[] { coffee,
50              throw new WebApplicationException(e, Response.Status.INTERNAL_SERVER_ERROR);
51          }
52      }
```

https://github.com/m-reza-rahman/weblogic-on-azure

Helidon – helidon.io

helidon.io

- API Standard
  - Helidon MP: MicroProfile
  - Helidon SE: Framework specific functional programming API
- Build system
  - Maven by default.
- Create app
  - $ mvn archetype:generate -DinteractiveMode=false \ -DarchetypeGroupId=io.helidon.archetypes \ -DarchetypeArtifactId=helidon-quickstart-mp \ -DarchetypeVersion=1.4.1 \ -DgroupId=io.helidon.examples \ -DartifactId=helidon-quickstart-mp \ -Dpackage=io.helidon.examples.quickstart.mp
- Run app locally
  - $ java -jar target/helidon-quickstart-mp.jar

Helidon Hello World

```
27  /**
28   * Simple Application that produces a greeting message.
29   */
30  @ApplicationScoped
31  @ApplicationPath("/")
32  public class GreetApplication extends Application {
33
34      @Override
35      public Set<Class<?>> getClasses() {
36          return CollectionsHelper.setOf(GreetResource.class);
37      }
38  }
39
```

Quarkus – quarkus.io

QUARKUS

- API Standard
  - MicroProfile / Java EE
- Build system
  - Maven by default. Supports Gradle.
- Create app
  - $ mvn io.quarkus:quarkus-maven-plugin:0.22.0:create \
    -DprojectGroupId=org.acme -DprojectArtifactId=getting-started \
    -DclassName="org.acme.quickstart.GreetingResource" -Dpath="/hello"
- Run app locally
  - $ mvn quarkus:dev

Quarkus HelloWorld

```
8   @Path("/")
9   public class Application {
10
11      @GET
12      @Produces(MediaType.TEXT_PLAIN)
13      public String hello() {
14          return "Hello App Service";
15      }
16
17  }
```

Spring Boot – start.spring.io

- API Standard
  - Proprietary, with dependencies on some Java EE APIs
- Build system
  - Maven or Gradle.
- Create app
  1. Visit start.spring.io
  2. Download the zip
- Run app locally
  - $ mvn quarkus:dev

No Framework – jdk.java.net

- API Standard
  - JDK Only
- Build system
  - Whatever you like
- Create app
  1. Visit package javadocs for com.sun.net.httpserver
  2. Copy source code
- Run app locally
  - $ java -jar myserver.jar

No Framework HelloWorld

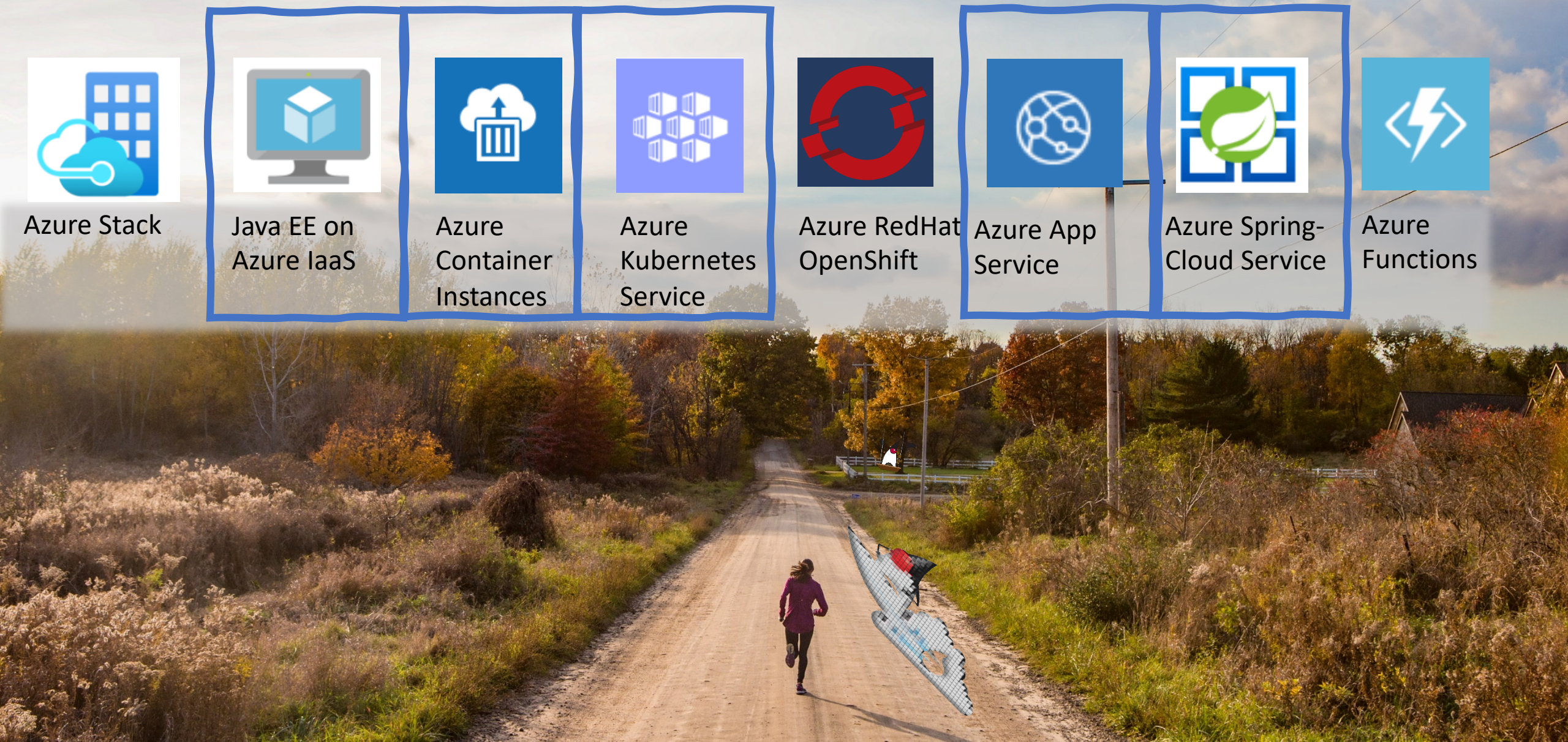
```java
public class HelloWorldServer {

    public static void main(String[] args) throws Exception {
        HttpServer server = HttpServer.create(new InetSocketAddress(8080), 0);
        server.createContext("/test", new MyHandler());
        server.setExecutor(null); // creates a default executor
        server.start();
    }

    static class MyHandler implements HttpHandler {
        @Override
        public void handle(HttpExchange t) throws IOException {
            String response = "Hello World";
            t.sendResponseHeaders(200, response.length());
            OutputStream os = t.getResponseBody();
            os.write(response.getBytes());
            os.close();
        }
    }

}
```

Java on Azure for Wherever You Are in your Cloud Journey



Java on Azure for Wherever You Are in your Cloud Journey

Java EE to Azure with Azure Virtual Machines

Azure Marketplace Solution for WebLogic

Published, maintained and supported by Oracle

Based on Oracle Linux 7.6, Oracle JDK and WebLogic 12c R2 (12.2.1.3)

Bring your own license

Supports common use-cases such as load-balancing and clustering

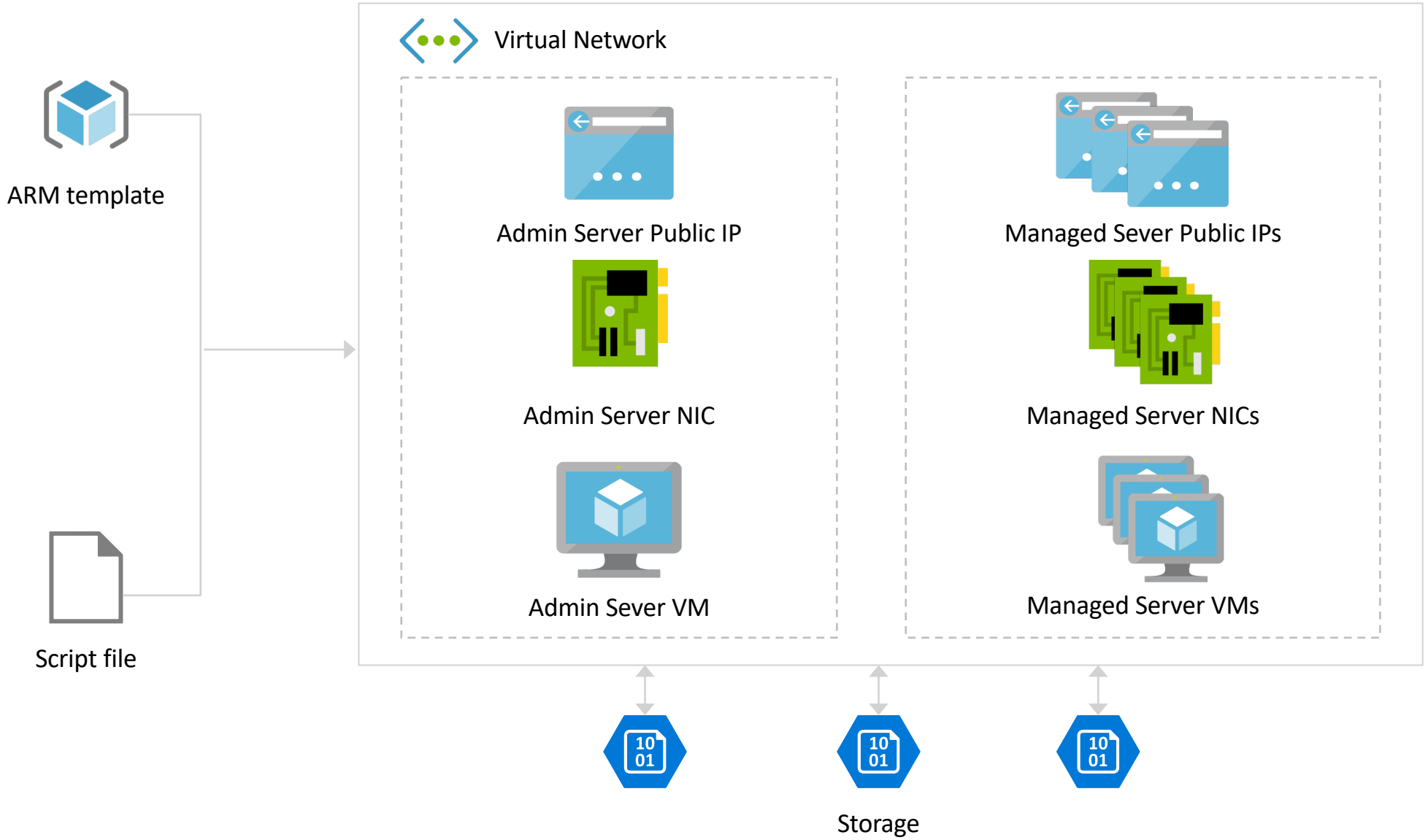
Simple to Start

- Single fully functional instance of WebLogic
- Admin only domain
- Starts admin server
- systemd configured for automatic restart

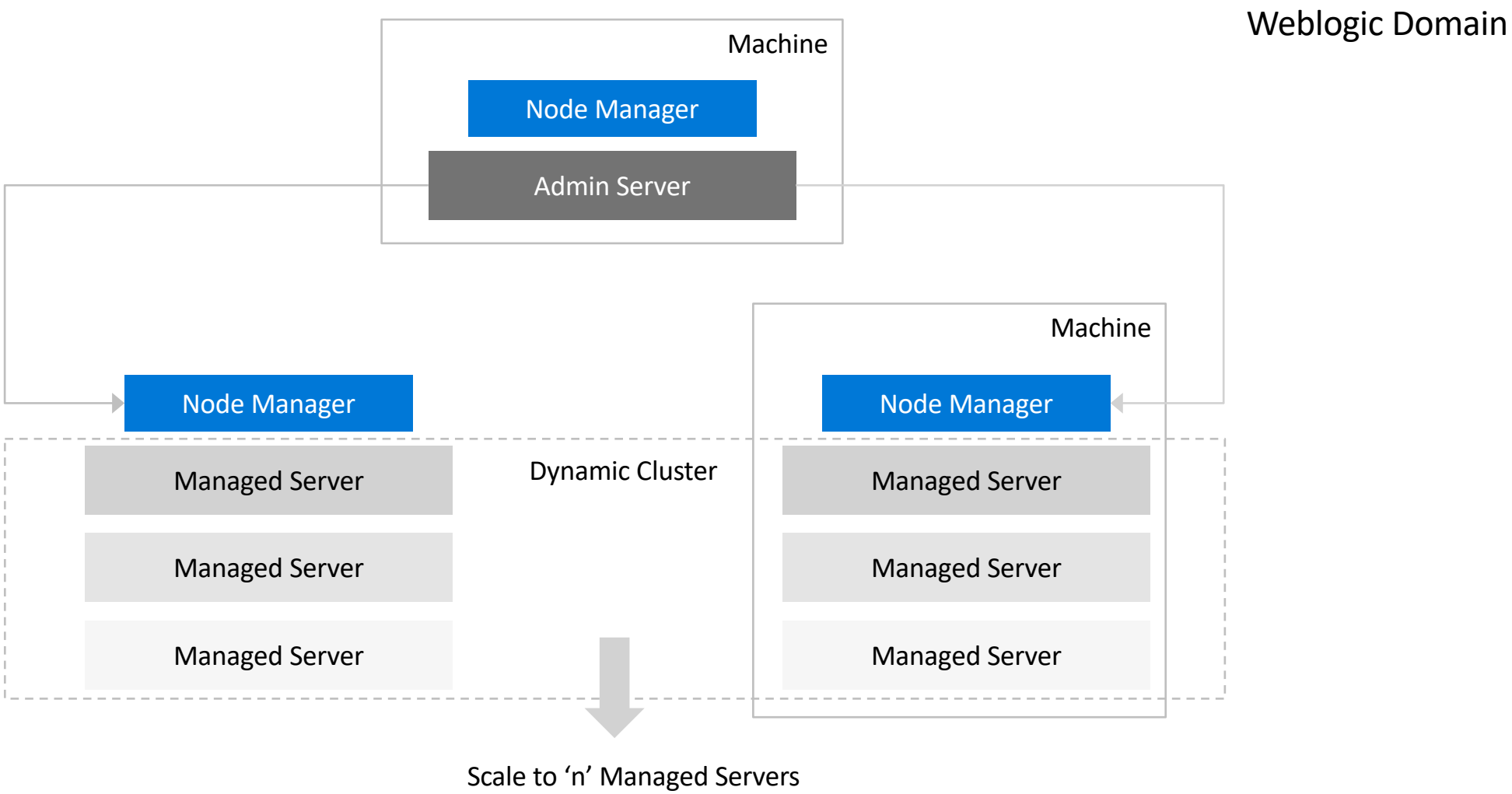
Clustering

- Fully functioning clustered multi-node domain
- User can specify number of nodes
- Admin server and all managed nodes are started by default when provisioning completes
- Admin server and node manager are started as systemctl service and CrashRecoveryEnabled is set to true for the node manager so even after a VM reboot servers are restarted automatically
- User can add nodes later
- Static and dynamic clustering

WebLogic Server on Azure IaaS



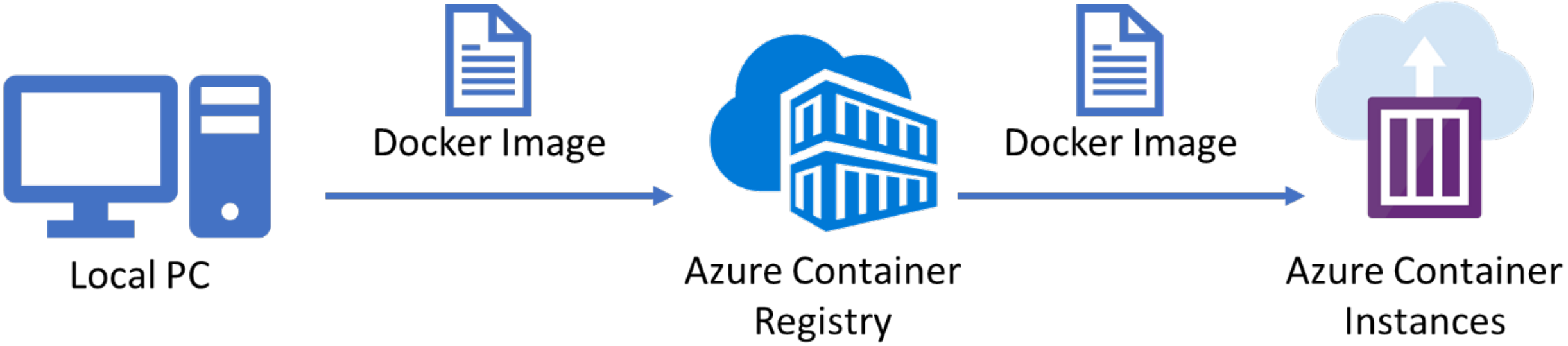
WebLogic Server on Azure IaaS



Java EE to Azure with Azure Virtual Machines

https://aka.ms/azurewls

Helidon to Azure with Azure Container Instances

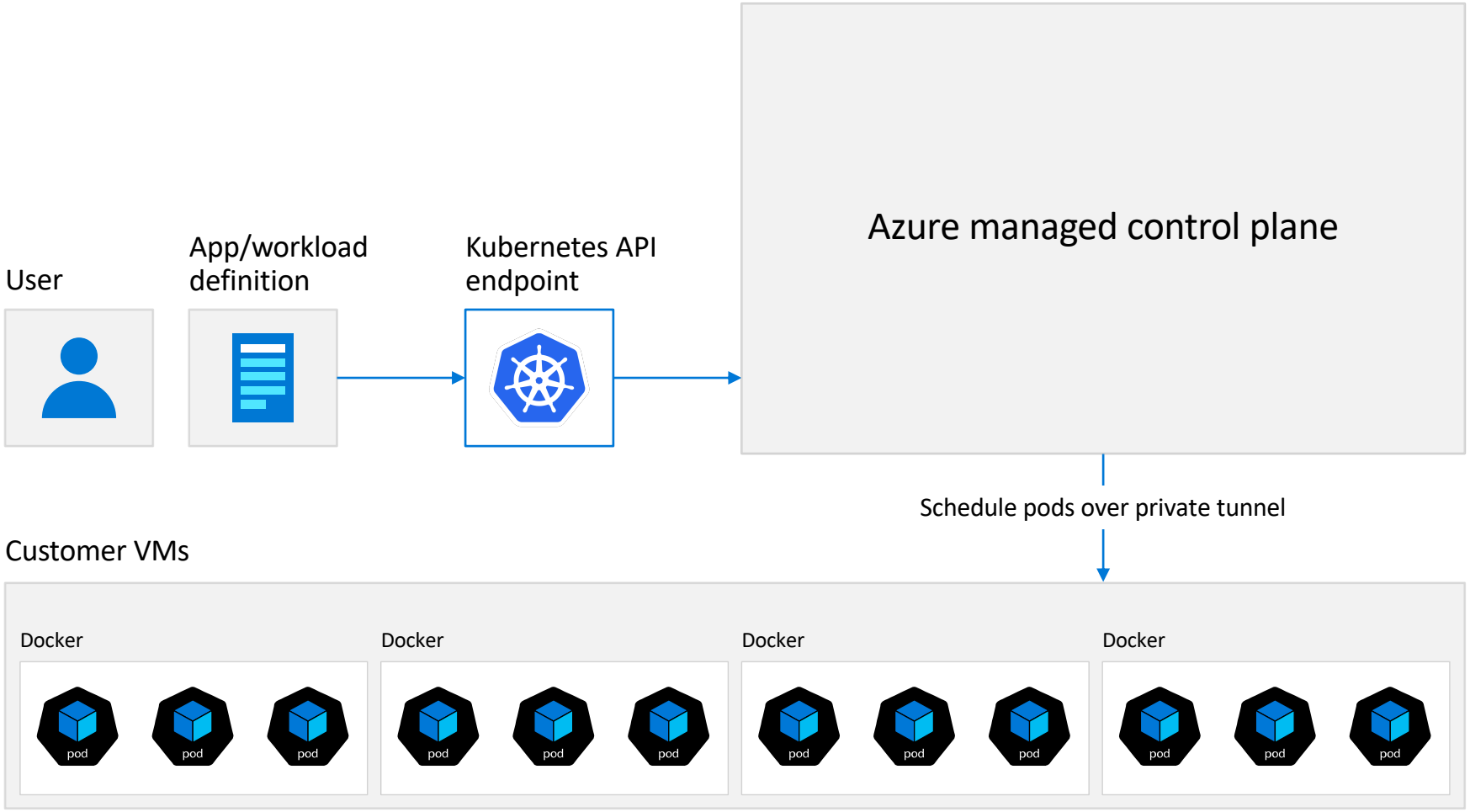


Helidon to Azure with Azure Container Instances

https://aka.ms/edburns/azurecontainer

Quarkus to Azure with Azure Kubernetes Service

- Automated upgrades, patches
- High reliability and availability
- Easy and secure cluster scaling
- Self-healing
- API server monitoring
- Control plane at no charge



Quarkus to Azure with Azure Kubernetes Service

https://aka.ms/edburns/azurek8s

Quarkus to Azure with Azure Kubernetes Service

1. Build the quarkus app locally
2. Use Graal to make a native executable
3. Use Docker to package the executable in a Docker image
4. Deploy image to Azure Container Registry
5. Create a Kubernetes cluster, bound to Azure Container Registry
6. Create Kubernetes `Deployment` declaring the app
7. Apply the `Deployment`
8. Hit the `LoadBalancer`

Azure Spring Cloud

A fully managed service for Spring Boot microservices

More choices and full integration into Azure's ecosystem and services

Fully managed Infrastructure

Built-in application lifecycle management

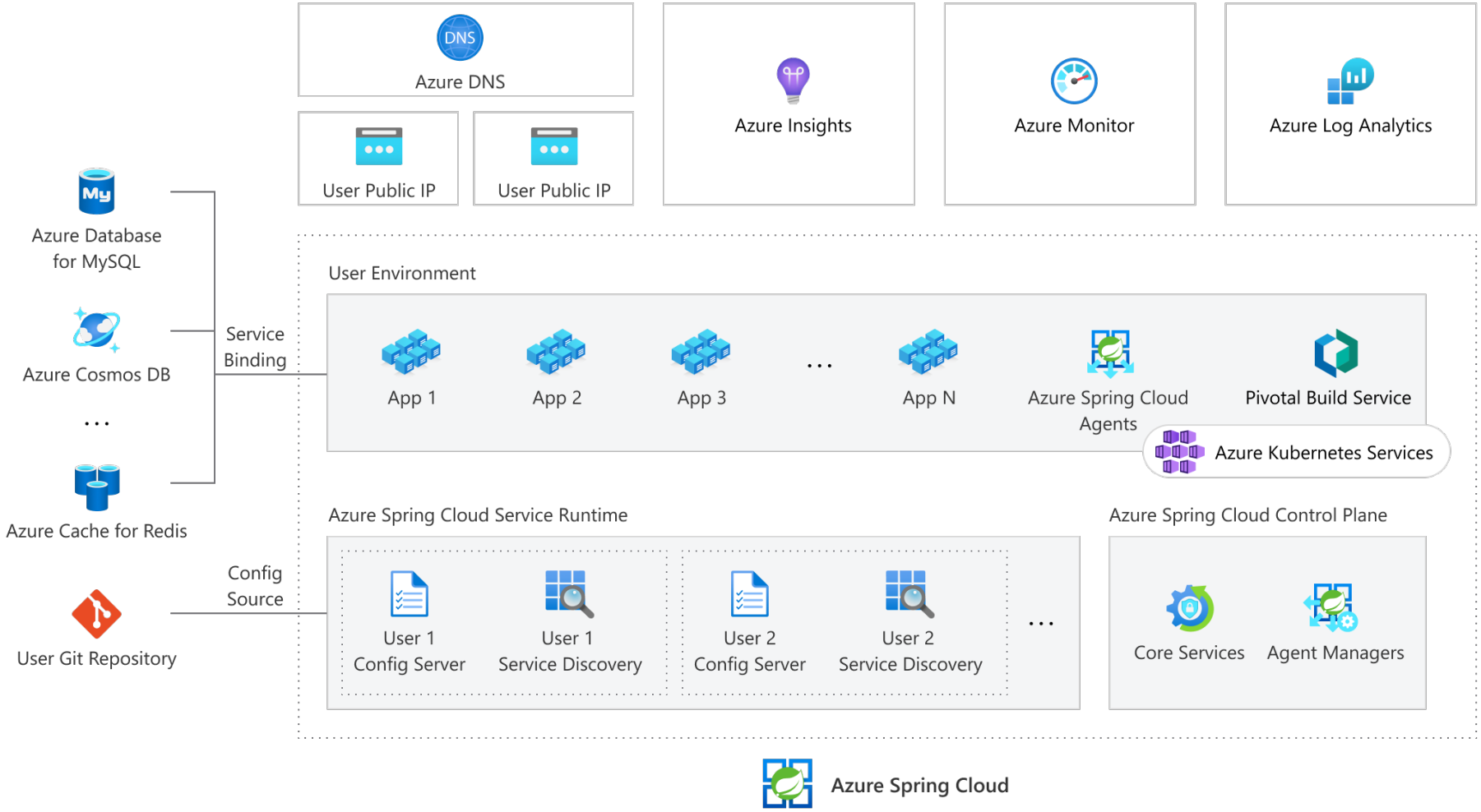
Ease of monitoring

Enterprise Ready

Spring Cloud - Architecture

Responsibilities	DIY with Spring Boot	Azure Spring Cloud Service
Application iteration, debugging	Customer	Customer
CI/CD	Customer	Microsoft
Build and manage Clusters	Customer	Microsoft
Host Spring Cloud Components	Customer	Microsoft
Monitoring and logging	Customer	Microsoft
Patching	Pivotal	Microsoft
Scaling	Customer	Microsoft
Support	Pivotal	Microsoft

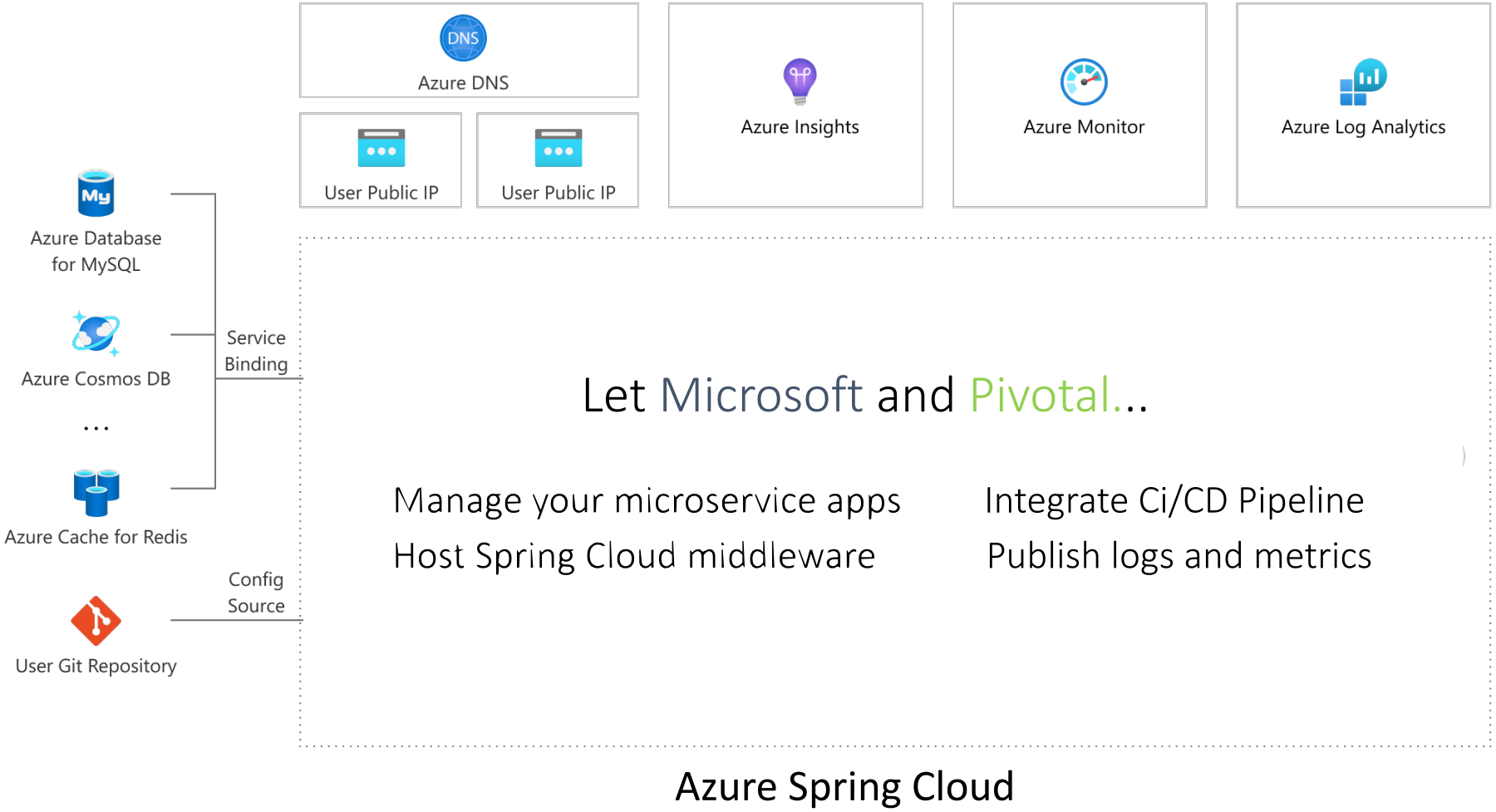
Customer Pivotal Microsoft



Simplify your cloud infrastructure for Spring boot applications

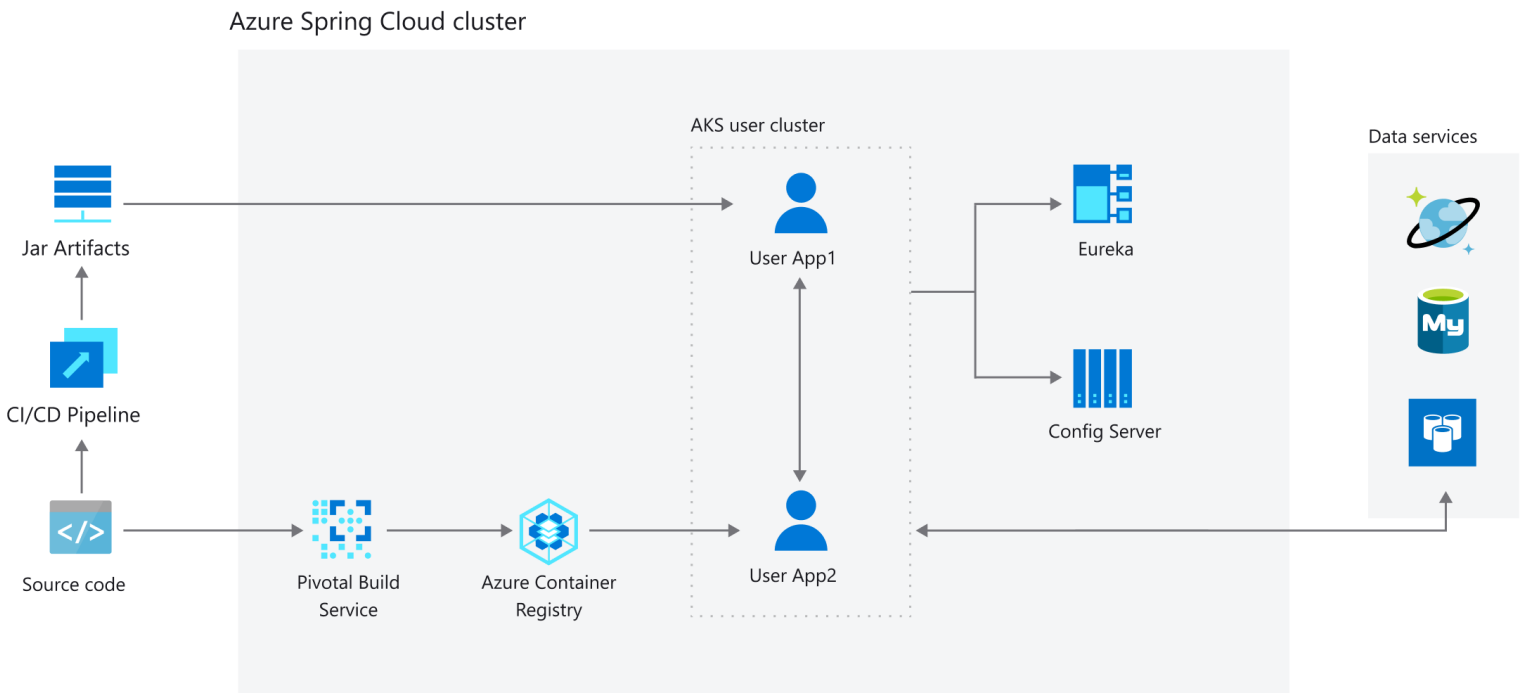
Responsibilities	DIY with Spring Boot	Azure Spring Cloud Service
Application iteration, debugging	Customer	Customer
CI/CD	Customer	Microsoft
Build and manage Clusters	Customer	Microsoft
Host Spring Cloud Components	Customer	Microsoft
Monitoring and logging	Customer	Microsoft
Patching	Pivotal	Microsoft
Scaling	Customer	Microsoft
Support	Pivotal	Microsoft

Customer Pivotal Microsoft



Built-in application lifecycle management

- Simple app lifecycle management
- Easily deploy source code or build artifacts
- Automatically wire your app with Spring Cloud infrastructure
- Integrated CI/CD pipeline for deployment

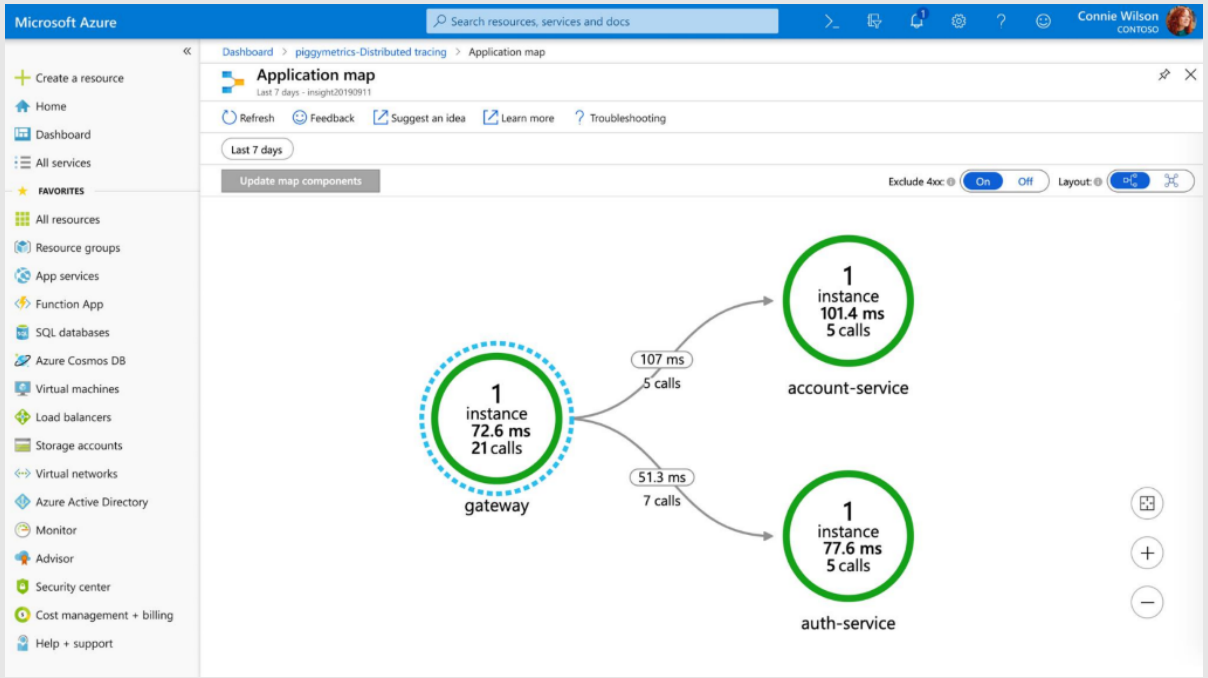


Monitor your apps

Gain insights with Azure monitor

Aggregate metrics

Identify reliability issues



Roadmap *(Calendar Year)*

Q3 2019

June
Limited Private Preview
- Application lifecycle management
- Config server
- Eureka
- Manual scaling

Q4 2019

October
Private Preview public announcement

- Additional capabilities in Config server
- Azure monitor
- Diagnostics
- Distributed tracing
- Build service
- Blue green deployment
- Service binding
- SSL
- UX revamp
- Documentation
- Deployment experience w/ Maven

November
Public Preview

- Full billing
- Customer support

Q1 2020

- Log streaming
- Alerts based on monitoring data
- Basic tier
- MSI support for Azure resources
- Custom domain
- Interactive self-diagnostics
- Auto patching systems and app runtime
- Jenkins integration
- VNET

Q2 2020

General Availability

- 99.9x SLA
- Available in more Azure Regions
- Production grade app monitor and diagnostics (e.g. app runtime inspection)
- Circuit breaker dashboard
- Auto scale
- E2E dev experience in IntelliJ

…
Other features based on customer feedback and market trends (e.g. Enterprise grade security)

Future

Candidates

- .NET core/Steeltoe
- mTLS among customers' applications
- Support certificate for outband traffic (e.g. from app to Data services)
- ….

Spring Boot to Azure with Azure Spring-Cloud Service

https://aka.ms/edburns/azurespringcloud

No Framework App to Azure with Azure App Service Linux

Web Apps that Scale with your Business

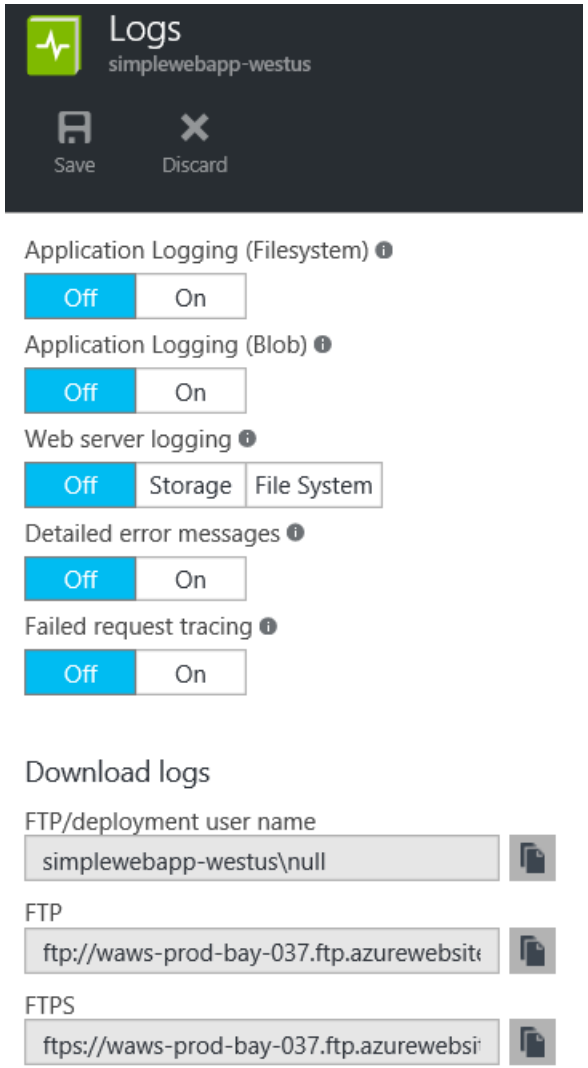
Full capability set available including:

- .NET, Node.js, Java, PHP, and Python
- WebJobs for long running tasks
- Integrated VS publish, remote debug…
- CI with GitHub, BitBucket, VSTS
- Auto-load balance, Autoscale, Geo DR
- Virtual networking and hybrid connections
- Site slots for staged deployments

App Service

Diagnostics Log Files

- Log File Targets
  - File System (Shared Cloud Drive)
  - Azure Blob Storage
- Web Server Log
  - File System or Azure Blob Storage
- Detailed Error Messages
- Failed Request Tracing
  - File System Only



App Service

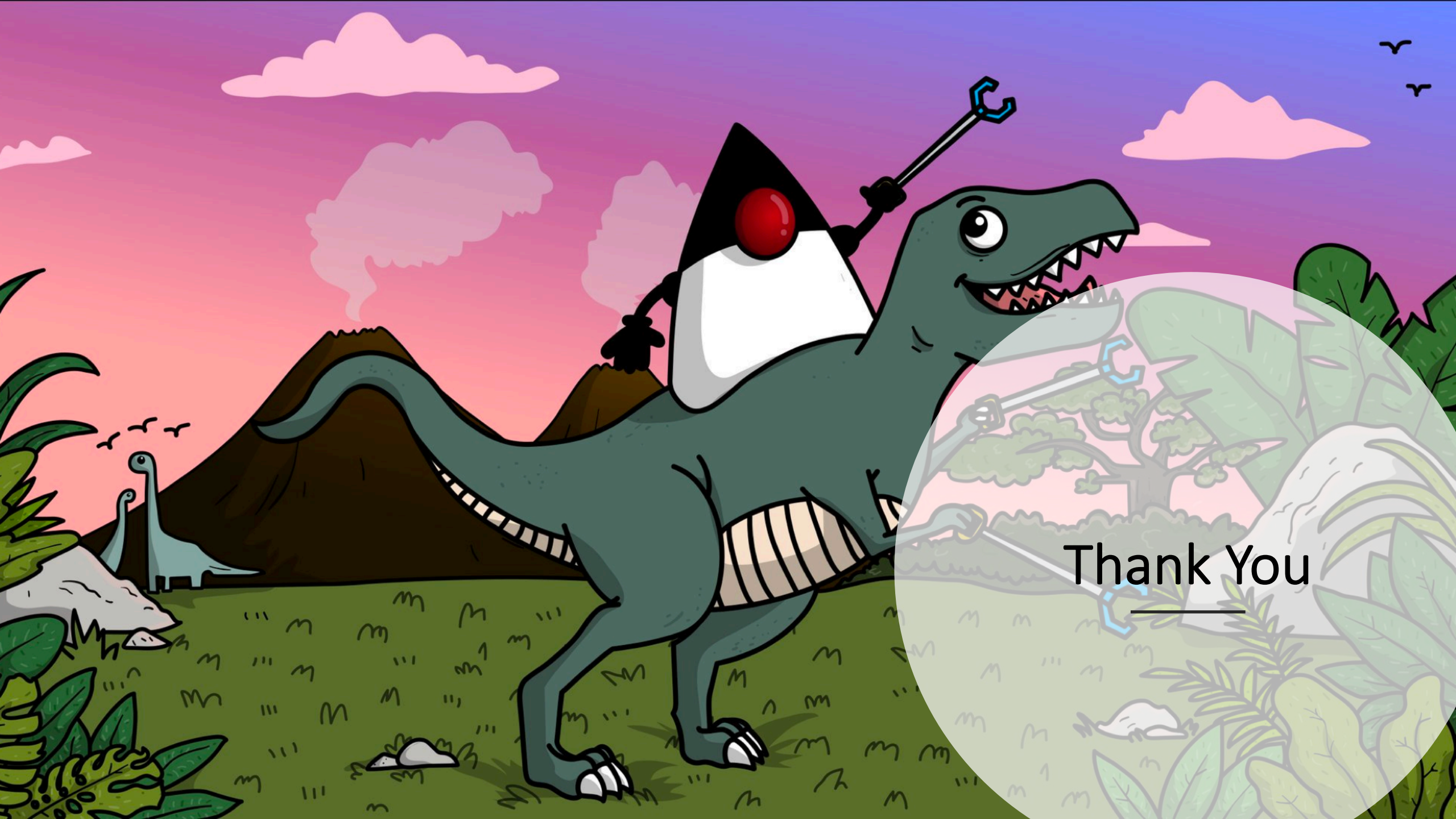
API Apps

- Easy API consumption
  - Integration with Swagger API metadata
  - Client SDK code generation (C#, Java, and JavaScript)

- Simple Access Control
  - Easily secure APIs using Azure AD or Social Logins (Facebook, Twitter, Google, Microsoft Account)
- Workflow Integration with Azure Logic Apps
- First Class Tooling Support
  - Visual Studio and Visual Studio Code
  - Maven

No Framework App to Azure with Azure App Service Linux

https://aka.ms/edburns/azureappservice



Thank You

Microsoft