



Azure Spring Cloud on Learn TV

Integrate Spring apps with Service Bus using
JMS 2.0

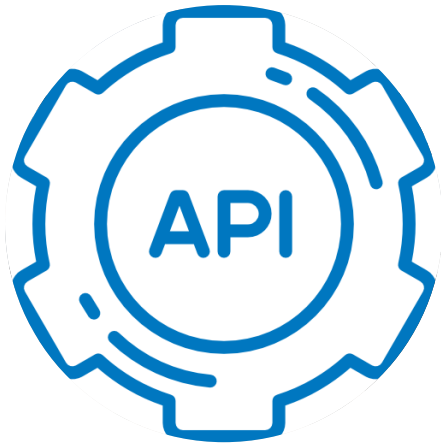
Ashish Chhabria
Program Manager, Azure Service Bus

Agenda

-
- What is Java Message Service (JMS) 2.0
 - How to use it?
 - Demo – Spring Petclinic microservices.
 - Benefits of using Azure Service Bus



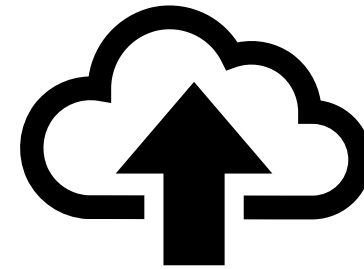
What is Java Message Service (JMS)?



Universal API for
Messaging



Code once
Run anywhere

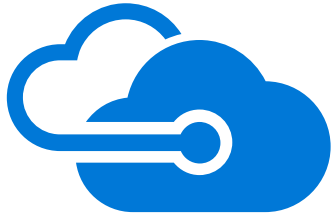


Seamlessly migrate
across messaging
brokers

Now available on Azure Service Bus Premium



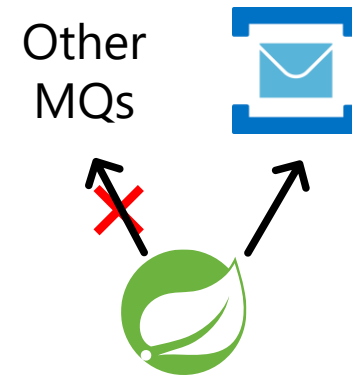
With Azure Service Bus, you now get...



Cloud scale
turnkey
messaging PaaS



Familiar JMS API
(introduced
August 1998)

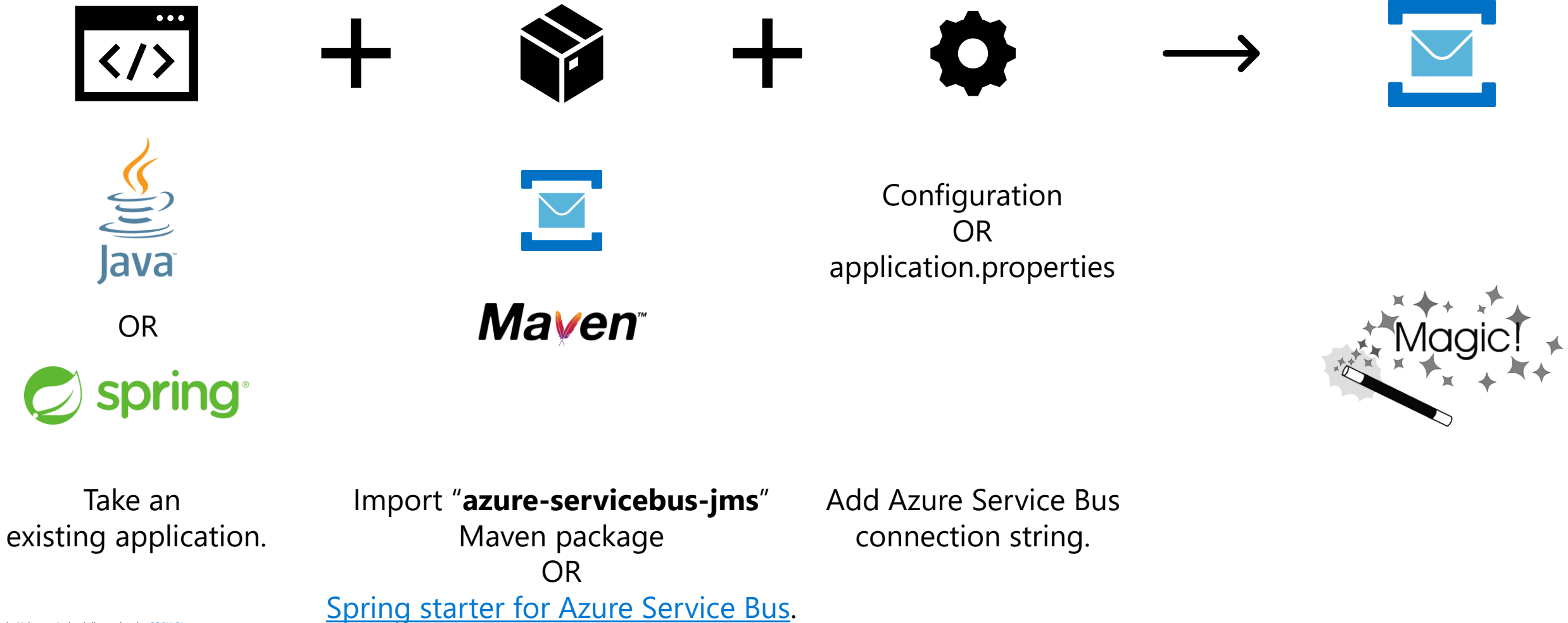


Spring Apps
seamlessly point
to Azure Service
Bus

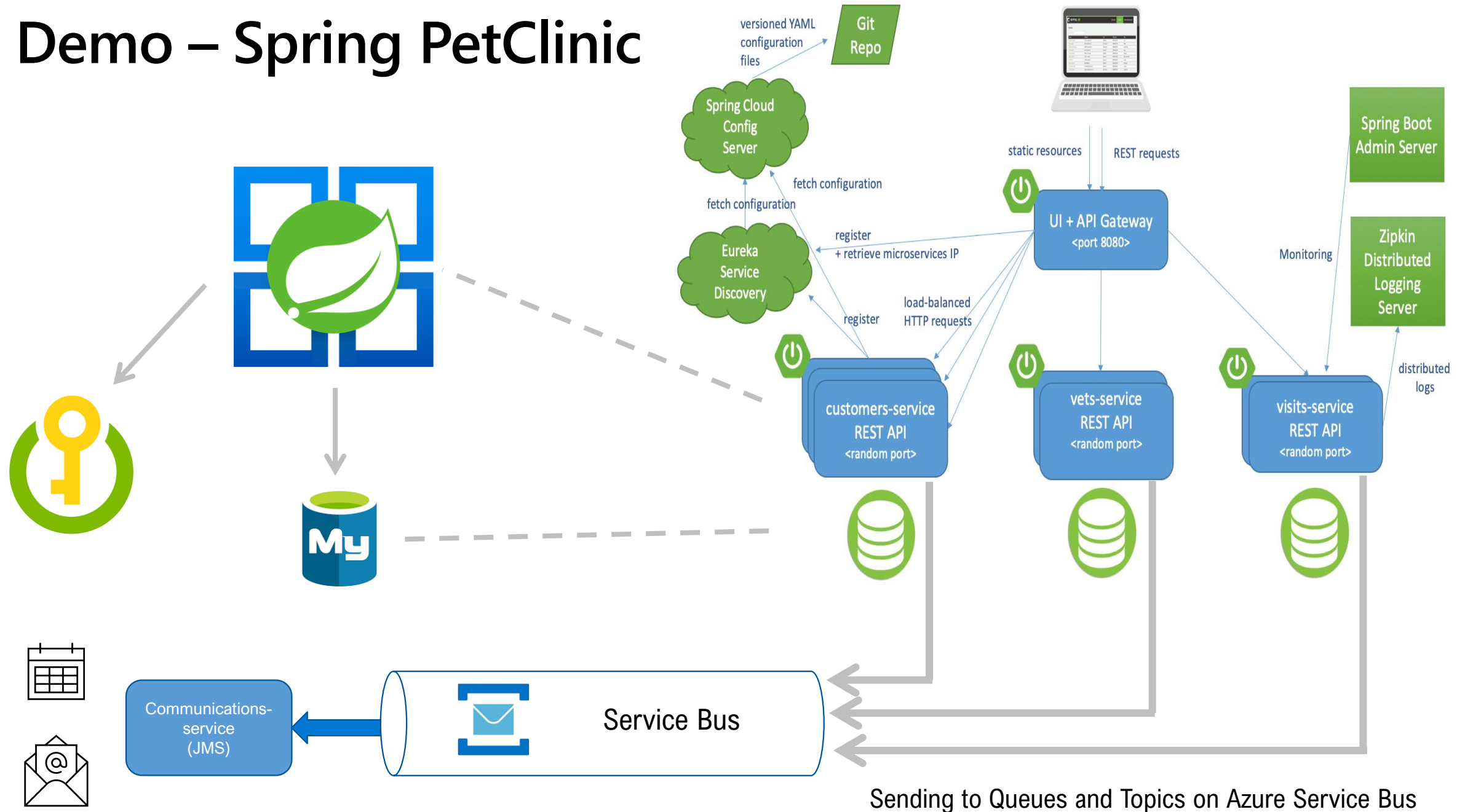


Deploy to Azure
Spring Cloud

That is cool – How can I use it?



Demo – Spring PetClinic



Spring PetClinic

Customers Service

```
public Owner createOwner(@Valid @RequestBody Owner
owner) {

    // Save pet owner info to database

    // Send pet owner info to Service Bus Queue
    jmsTemplate.convertAndSend(DESTINATION_NAME,
objMapper.writeValueAsString(savedOwner));

    // Return pet owner info
    return savedOwner;
}
```

Receive from Queue (JMSListener)

```
@JmsListener(destination = QUEUE_NAME,
containerFactory =
"jmsListenerContainerFactory")
public void receiveMessage(String ownerString)
throws IOException, InterruptedException {

    // Retrieve pet owner information
    String refinedOwnerString =
ownerString.substring(ownerString.indexOf("{")
);

    // Send email to pet owner
emailService.sendOwnerWelcomeEmail(refinedOwnerString);

}
```

Spring PetClinic

Communications Service

Spring PetClinic

Visits Service

```
Visit create(  
    @Valid @RequestBody Visit visit,  
    @PathVariable("petId") int petId) {  
  
    // Save visit information  
  
    // Send visit information to topic  
  
    jmsTemplate.convertAndSend(topic,  
        objectMapper.writeValueAsString(savedVisit));  
  
    // Return visit information  
    return savedVisit;  
}
```

Receive from Topic using Subscription (JMSListener)

```
@JmsListener(destination = TOPIC_NAME,  
subscription = "visitRecordSubscriber",  
containerFactory =  
"topicJmsListenerContainerFactory")  
public void receiveMessage(String  
visitRecordString) throws IOException {  
    // Retrieve Visit information  
    String refinedVisitRecordString =  
        visitRecordString.substring(visitRecordString.  
            indexOf("{"));  
  
    // Send email to pet owner about recent visit  
    emailService.sendVisitRecordEmail(refinedVisit  
        RecordString);  
}
```

Spring PetClinic

Communications Service

Spring PetClinic

Communications Service

```
public List<OwnerDetails> index() {  
  
    List<OwnerDetails> owners =  
        jmsTemplate.browse(queueName, new  
            BrowserCallback<List<OwnerDetails>>() {  
  
                // Retrieve browsed messages and add to list of  
                owners  
  
            });  
}
```

Spring PetClinic

Configuration

pom.xml +
application.properties.

Pom.xml

```
<dependency>  
    <groupId>com.microsoft.azure</groupId>  
    <artifactId>azure-servicebus-jms-spring-boot-  
starter</artifactId>  
</dependency>
```

Application.properties

```
spring.jms.servicebus.connection-string=<ServiceBusNamespaceConnectionString>  
spring.jms.servicebus.idle-timeout=<IdleTimeout>
```

Benefits of using Azure Service Bus ...



Fully managed PaaS

- No more VMs.
- First class High availability and Disaster Recovery.
- Simple pricing.
- Integrate with Azure PaaS.



Monitor and AutoScale

- Setup alerts using Azure Monitor.
- Scale up/down to adapt to change in workloads.

Secure by design

- Bring your own key (BYOK) encryption at rest.
- Integration with Azure AD.
- Secure access with Virtual Networks, Firewalls and Private endpoints.

Use JMS

- Seamless migration from other JMS providers.

Azure Service Bus Resources



- Get started with JMS 2.0 on Azure Service Bus : aka.ms/jmsGettingStarted
- Migrate apps using ActiveMQ over JMS 2.0 to Azure Service Bus : aka.ms/jmsActiveMQToServiceBus
- Learn more about Azure Service Bus: azure.microsoft.com/services/service-bus/
- For feedback and questions, please reach out to askservicebus@microsoft.com
- Demo – Spring Petclinic - github.com/axisc/spring-petclinic-microservices/

Get started – build your cloud-native solutions today!

- Get started with Azure Spring Cloud using quickstart: aka.ms/azure-spring-cloud-start
- Learn using a self-paced workshop on GitHub: aka.ms/azure-spring-cloud-github
- Learn about implementing solutions on Azure Spring Cloud: aka.ms/azure-spring-cloud-docs
- Migrate your apps Azure Spring Cloud
 - Spring Boot: aka.ms/azure-spring-cloud-migrate-springboot
 - Spring Cloud: aka.ms/azure-spring-cloud-migrate-springcloud
 - Tomcat: aka.ms/azure-spring-cloud-migrate-tomcat
- Wire Spring apps to interact with Azure services: aka.ms/spring-integrations
- For feedback and questions, please reach out to spring-team@microsoft.com



LEARN

Microsoft.com/Learn