

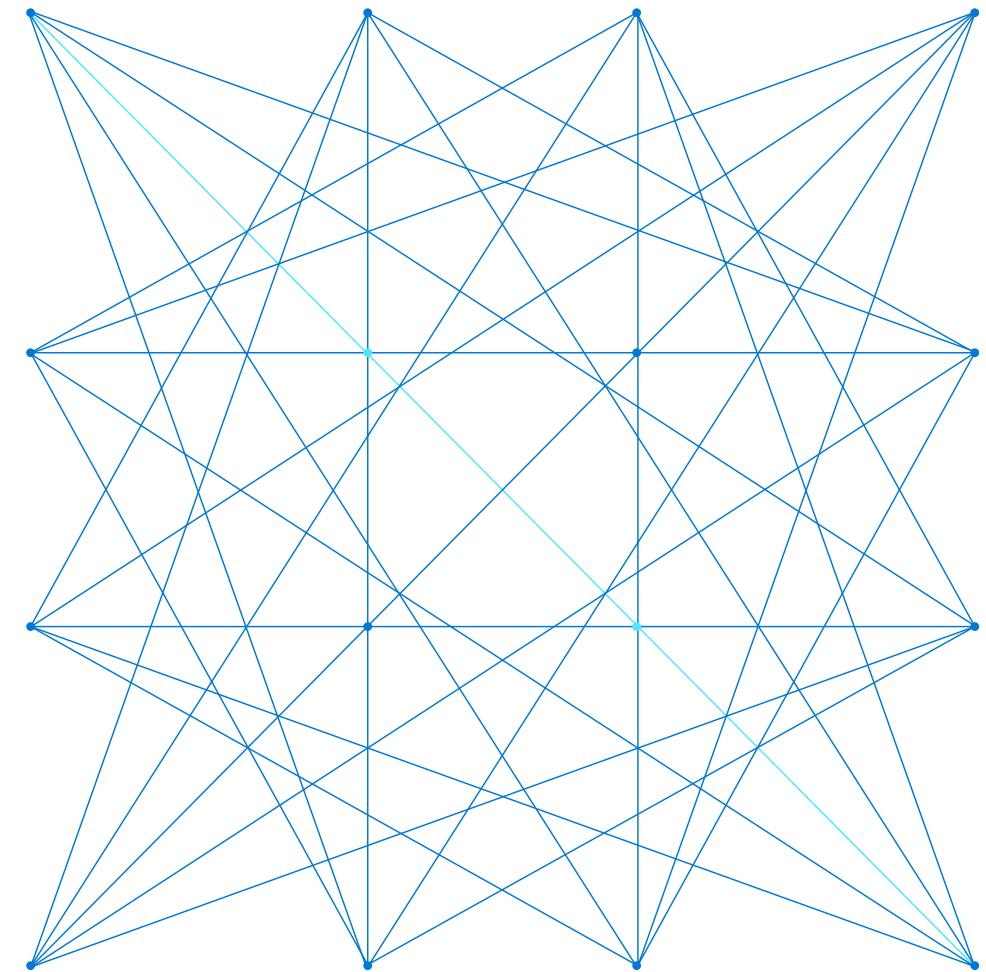


The Java Developer Experience on Azure

@RoryPreddy



Microsoft ❤️ Java



Java is widely used across Microsoft

500,000+ JVMs in production*



LinkedIn

- Over 1800+ Java microservices in production
- Over 60+ Java open-source projects on GitHub



Mojang's Minecraft

- Hundreds of servers built in Java
- Client Java Edition is very popular



Azure

- Services dependent on Java include, Azure HDInsight, Azure Databricks, and Azure Spring Cloud, and internal control systems



Android

- 50+ apps published by Microsoft in Google Play Store
- New Surface Duo phone based on Android OS



Yammer

- Most of back-end implemented in Java

*Internal usage; does not include customer workloads

Strong partner ecosystem



Jointly built and operated service—Azure Spring Cloud



Redis Enterprise on Azure managed infrastructure



Jointly supported JBoss EAP on App Service* and VMs



Elastic Cloud on Azure Marketplace



Solution templates for WebLogic on Azure IaaS



Confluent Cloud with Azure Portal and CLI integration



IBM WebSphere / WebSphere Liberty on Azure



OpenJDK support for all LTS Java versions

Community engagement

1,500,000+ VS Code Java Users



Toolkit and plug-ins to improve
Eclipse dev experience



3.6M+ Java repositories hosted, free
support and resources for Java devs



MS Build of OpenJDK, fixes,
enhancements, and new ports



Visual Studio Code

Java extensions for Maven, Run/
Debug, Unit Testing, and more



Founding member and
project contributor

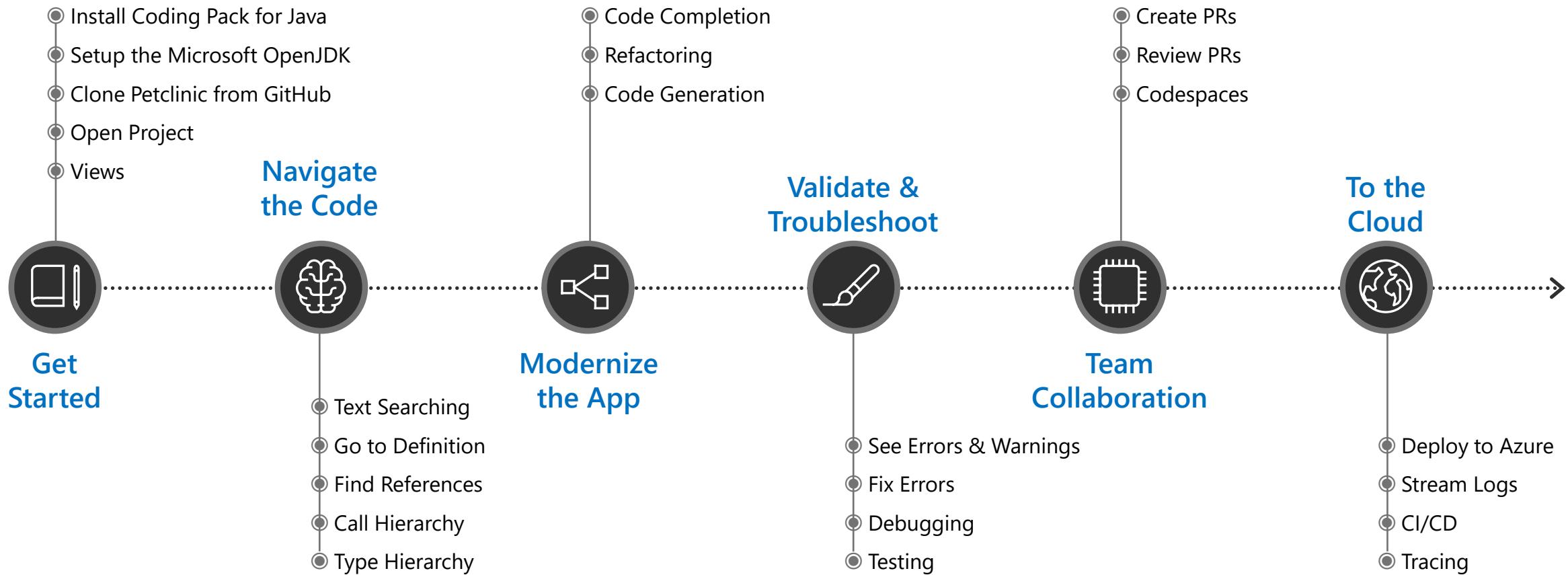


Support for popular Java
standards and projects



Spring Cloud for Azure project
and managed service for
Spring Boot

Capabilities



Get Started (Installation)



Download & install

<https://www.microsoft.com/openjdk>



Download & install

<https://code.visualstudio.com/>



Launch VS Code & install

Extension pack for Java from the
marketplace

OR



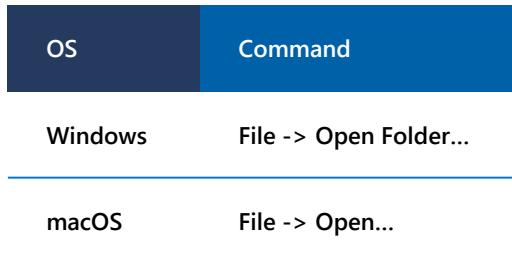
<https://aka.ms/coding-pack>

* If you only have JDK 8 or earlier installed, **you need to install a JDK 11+** side by side to run the tools

Get Started (Load Projects)

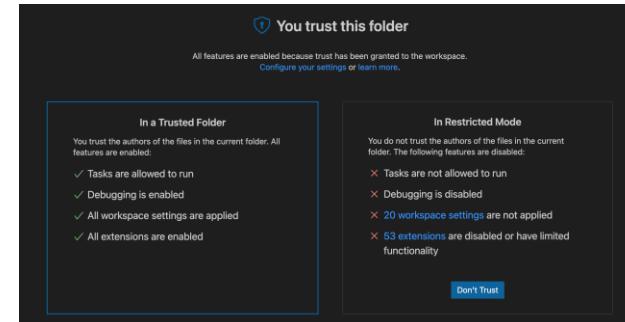
1

Load projects by **Opening Folders**



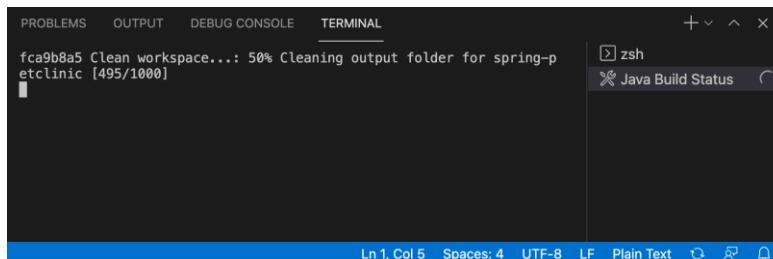
2

Workspace Trust is needed to enable all features. Please only trust the folder you cloned from a trustworthy source



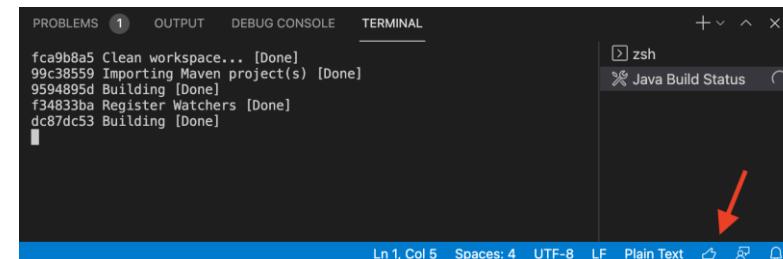
3

You can **View Build Status** by clicking the spinning icon on status bar



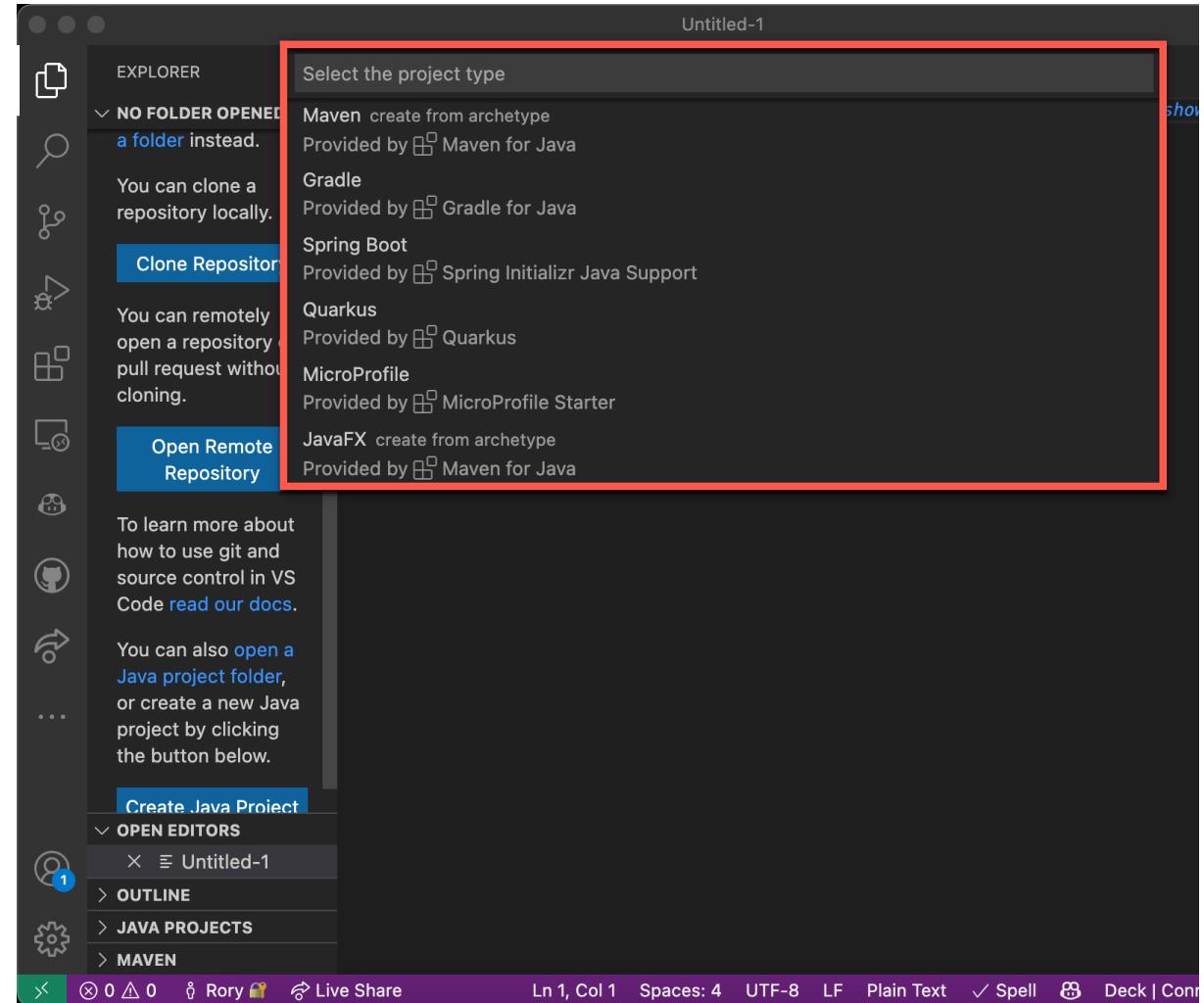
4

All features are **ready** when the project are fully imported



Get Started – Build Systems

- Maven/Gradle archetypes
- Spring/Quarkus/Microprofile starters
- JavaFX



Get Started - Project Explorer

- Inspect your packages
- Run/debug applications
- Export JARs

The screenshot shows the VS Code interface with the following details:

- Project Explorer (Left):** Shows the project structure under "JAVA PROJECTS". A red box highlights the "spring-petclinic" package, which contains "src/main/java", "src/main/resources", "src/test/java", and "target/generated-sources/annotations". It also lists "JRE System Library [JavaSE-1.8]". Other packages shown are "org.springframework.samples.petclinic.m..." and "org.springframework.samples.petclinic.o...".
- Editor (Right):** Displays the file "OwnerController.java" with code related to Spring Boot and Java annotations.
- Bottom Status Bar:** Shows the current branch as "master*", file count as "0 △ 1", and line count as "1".
- Bottom Activity Bar:** Includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL.

Inspect the Code

- **Right-click** on the symbols to Navigate
- **Hover** to show Javadocs
- **Ctrl/Command + Click** to Go to Definition

The screenshot shows a code editor window for `OwnerController.java`. A context menu is open over the line of code:

```
    if (result.hasErrors()) {  
        return VIEWs_OWNER_CREATE_OR_UPDATE_FORM;  
    } else {  
        this.owners.add(owner);  
        return "redirect:/owners";  
    }  
}
```

The context menu, which is highlighted with a red box, contains the following options:

- Go to Definition (F12)
- Go to Type Definition
- Go to Implementations (⌘ F12)
- Go to References (⇧ F12)
- Go to Super Implementation
- Peek >
- Find All References (⌃ ⌘ F12)
- Find All Implementations
- Show Call Hierarchy (⌃ ⌘ H)
- Show Type Hierarchy
- Rename Symbol (F2)
- Change All Occurrences (⌘ F2)
- Format Document (⌃ ⌘ F)
- Format Document With...
- Refactor... (⌃ ⌘ R)
- Source Action...
- Commit Changes >
- Cut (⌘ X)
- Copy (⌘ C)
- Copy As >
- Paste (⌘ V)

The status bar at the bottom indicates:

PROBLEMS 1 OUTPUT DEBUG
overwritten by merge:
 .vscode/settings.json
 pom.xml
Please commit your changes or
Abortin
roml@Romes-MacBook-Pro-16 spring

Faisal Hameed, 5 years ago via PR #152 Ln 67, C

Code Editing – Auto-completion

- **Type** to see suggestions
- **Ctrl + Space** to show suggestions

The screenshot shows a code editor window with the file `OwnerController.java` open. The code is part of a Spring application, specifically handling owner search logic. A red box highlights the auto-completion dropdown menu that appears when the user types the word `for`. The dropdown lists several Java constructs related to iteration:

- `for`
- `foreach` (selected)
- `fori`
- `ForAbstractMethod`
- `ForAdvice`
- `ForAdvice - net.bytebuddy.asm.Advice.ArgumentHandler`
- `ForAdvice - net.bytebuddy.asm.Advice.MethodSizeHandle...`
- `ForAdvice - net.bytebuddy.asm.Advice.StackMapFrameHa...`
- `ForAllArguments`
- `ForAnnotation`
- `ForAnnotationDescription`
- `ForAnnotationProperty`

The code editor interface includes a status bar at the bottom showing the current file path (`OwnerController.java`), line number (91), column number (12), and encoding (UTF-8). There are also tabs for Java and JavaSE-1.8.

Code Editing - Refactoring

- **Right-click -> Refactor... to show refactoring options**
- **F2 to Rename**

The screenshot shows a code editor window for `OwnerController.java`. The file contains Java code for a Spring MVC controller. A context menu is open over the word `owner` in the line `model.put("owner", new Owner());`. The menu has several options:

- Go to Definition
- Go to Type Definition
- Go to Implementations
- Go to References
- Go to Super Implementation
- Peek
- Find All References
- Find All Implementations
- Show Call Hierarchy
- Show Type Hierarchy
- Rename Symbol** (highlighted with a red box)
- Change All Occurrences
- Format Document
- Format Document With...
- Format Selection
- Refactor...** (highlighted with a blue box)
- Source Action...
- Commit Changes
- Cut
- Copy
- Copy As
- Paste

The code editor interface includes a sidebar with navigation links like `clinic`, `Controller`, `Repository`, and `Resource`. The bottom status bar shows the file path `OwnerController.java — spring-petclinic` and the line number `Ln 87, C`.

Code Editing – Code Generation

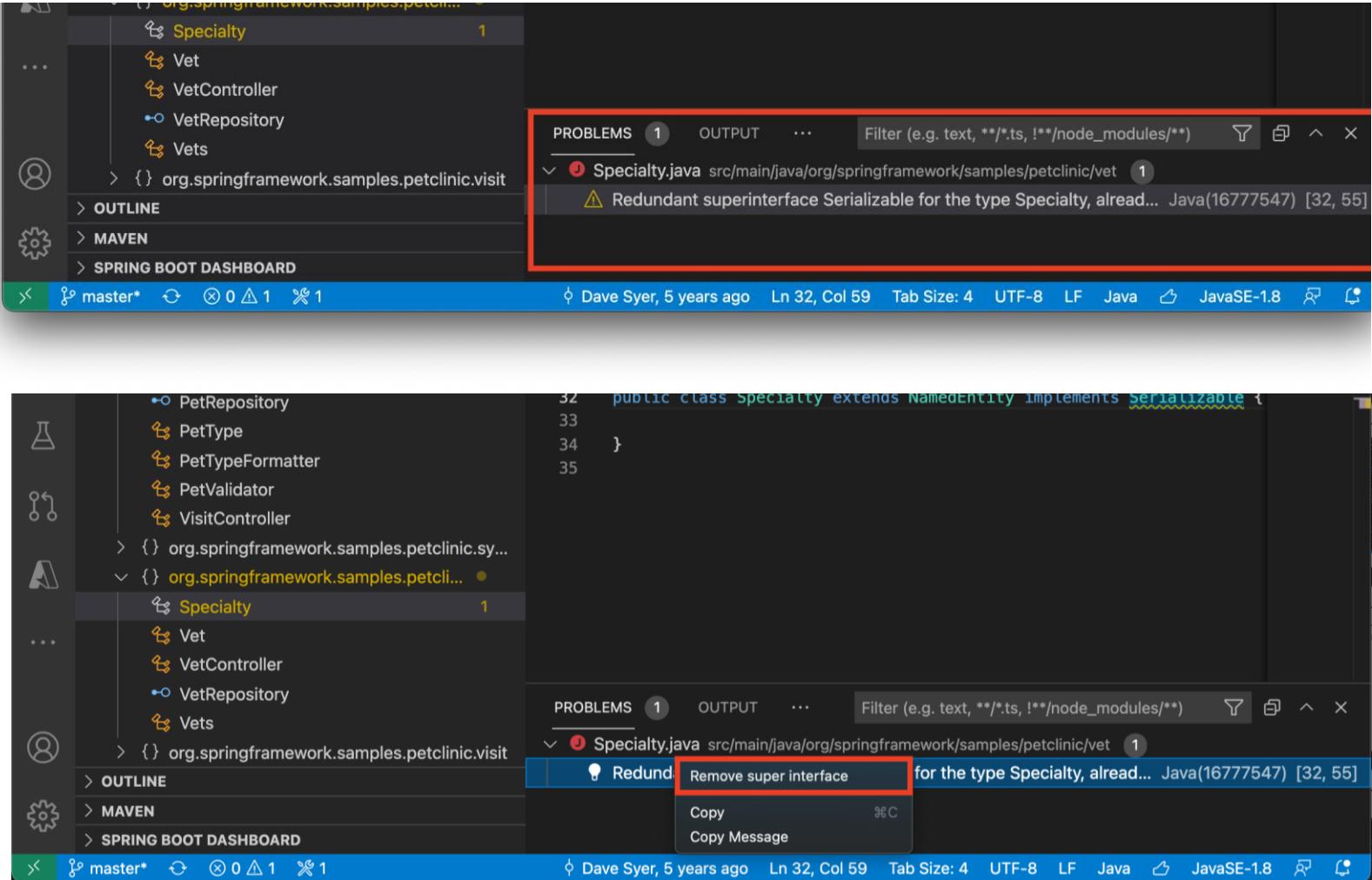
- **Right-click -> Source Action...** to see Code Generation options
- **Option + Shift + O** to **Organize Imports**
- Clicking on the **Code Action** lightbulb for Quick Fixes.

The screenshot shows the VS Code interface with the following details:

- File Structure:** On the left, there's a tree view showing project files like 'Controller', 'Repository', and 'Service' under 'petclinic'.
- Code Editor:** The main editor area contains Java code for 'OwnerController.java'. A context menu is open over the code at line 84.
- Context Menu Options:** The menu includes options like 'Generate Tests...', 'Organize imports', 'Override/Implement Methods...', etc., followed by a separator and 'Source Action...' which is highlighted with a red box.
- Bottom Status Bar:** Shows 'em Library [JavaSE-1.8]', 'overwritten by merge: .vscode/settings.json pom.xml', 'Please commit your changes or Aborting', 'roml@Romes-MacBook-Pro-16 sprin', '0 △ 1 ✘ 1', and 'Gordon Dickens, 8 years ago via PR #2 Ln 87'.

Validate & Troubleshoot – See Errors & Warnings

- Go to **Problems** view to check build errors & warnings
- **Right-click on errors** to see quick-fixes



Validate & Troubleshoot – Fix Errors

- **Click on the lightbulb** to see Quick-fix options

OwnerController.java — spring-petclinic

The screenshot shows the `OwnerController.java` file in VS Code. A red arrow points from the text "Quick-fix options" in the previous slide to a yellow lightbulb icon located at line 48, column 10. A tooltip menu is open, listing several quick-fix options: Extract to field, Extract to method, Extract to local variable (replace all occurrences), Extract to local variable, and Introduce Parameter... . The code editor shows imports for `OwnerRepository`, `ClinicService`, and `VisitRepository`. The `OwnerController` class has methods for initializing creation forms and handling owner creation or update.

```
* @author Michael Isvy
*/
@Controller
class OwnerController {

    private static final String VIEWS_OWNER_CREATE_OR_UPDATE_FORM = "owner/createOrUpdate主人";
    private final OwnerRepository owners;
    private VisitRepository visits;

    public OwnerController(OwnerRepository clinicService) {
        this.owners = clinicService;
    }

    @GetMapping("/owners/new")
    public String initCreationForm(Map<String, Object> model) {
        Owner owner = new Owner();
        model.put("owner", owner);
        return VIEWS_OWNER_CREATE_OR_UPDATE_FORM;
    }
}
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL COMMENTS

overwritten by merge:
 .vscode/settings.json
 pom.xml
Please commit your changes or stash them before you merge.
Aborting
roml@Romes-MacBook-Pro-16 spring-petclinic %

0 △ 1 ✎ 1

Dave Syer, 5 years ago Ln 48, Col 10 Spaces: 4 UTF-8 LF Java

Validate & Troubleshoot – Debugging

- **F5** to start debugging
- Use **DEBUG AND RUN** viewlet to inspect variables, callstacks and breakpoints
- Use the **Debug Toolbar** to control the execution

The screenshot shows the IntelliJ IDEA interface during a debug session. The title bar indicates the file is `OwnerController.java` from the `spring-petclinic` project. The `RUN AND DEBUG` toolbar at the top right includes icons for pausing, resuming, stopping, and exiting the debugger.

The `VARIABLES` viewlet on the left displays the `Local` variables for the current thread, which include `owner`, `result`, `model`, and `this`. The `WATCH` section is currently empty.

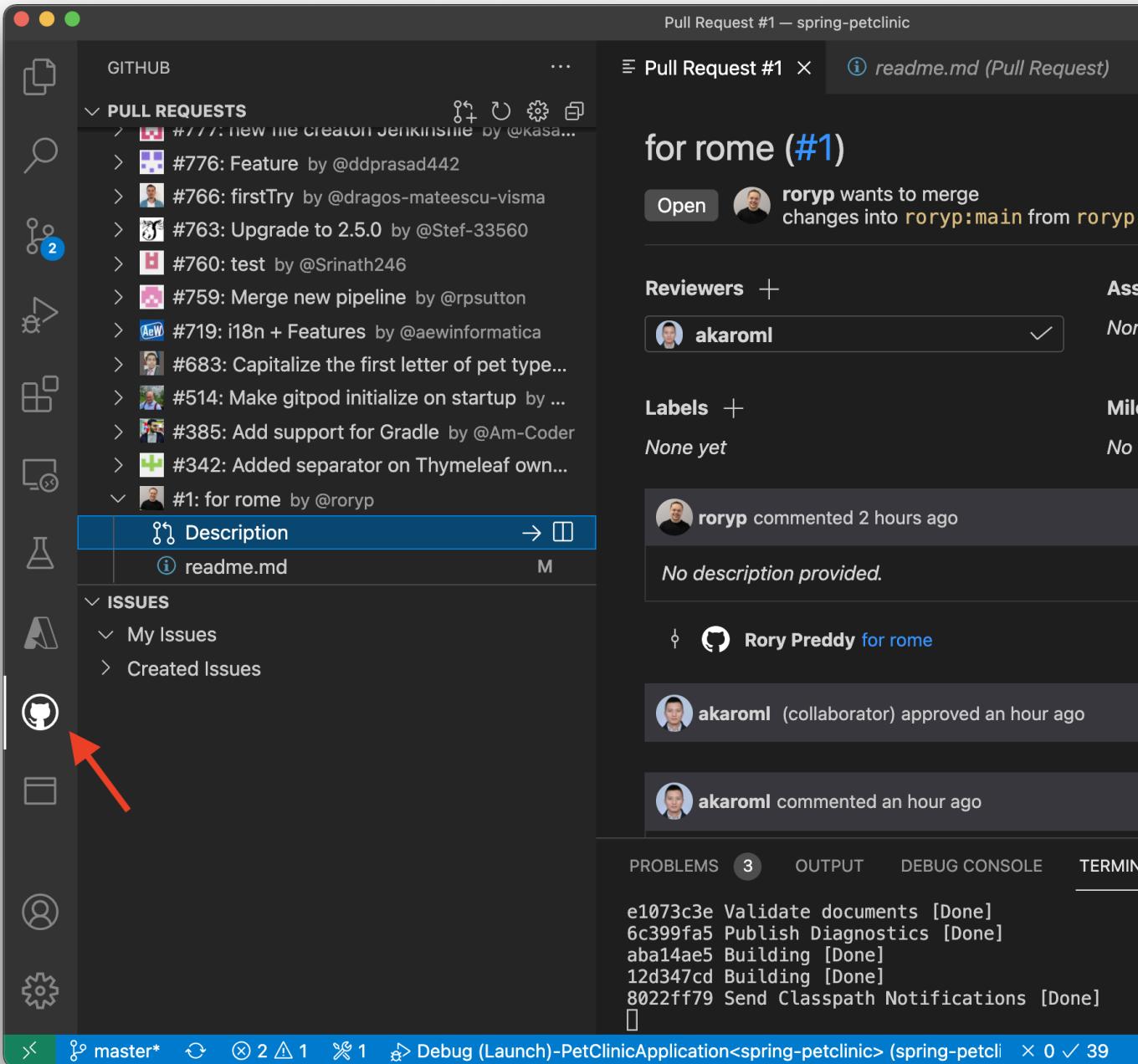
The `CALL STACK` section shows the current stack trace. The top frame is `Thread [http-nio-8080-exec-2] RUNNING`. Below it, a frame for `Thread [http-nio-8080-...] PAUSED ON BREAKPOINT` is expanded, showing the method `OwnerController.processFindForm(Owner, BindingResult)` and its call to `NativeMethodAccessorImpl.invoke0(Method, Object)`. The `BREAKPOINTS` section at the bottom lists several breakpoints that are currently active (indicated by checked checkboxes). The breakpoints include `AlreadyBoundException.class` at line 49, `AppendingBatchVisitor.class` at line 57, and `OwnerController.java` at lines 74 and 59.

The code editor on the right shows the `processFindForm` method. Line 84 is highlighted with a yellow background, and the code reads: `if (owner.getLastName() == null) { owner.setLastName(""); // empty string signifies broadest search } // find owners by last name Collection<Owner> results = this.owners.findByLastName(owner.getLastName()); if (results.isEmpty()) { // no owners found result.rejectValue("lastName", "notFound", "not found"); return "owners/findOwners"; } else if (results.size() == 1) { // 1 owner found owner = results.iterator().next(); }`

The `DEBUG CONSOLE` at the bottom shows some log output, including a warning about DevTools properties and a timestamp of `2021-07-15 17:24:44.840`.

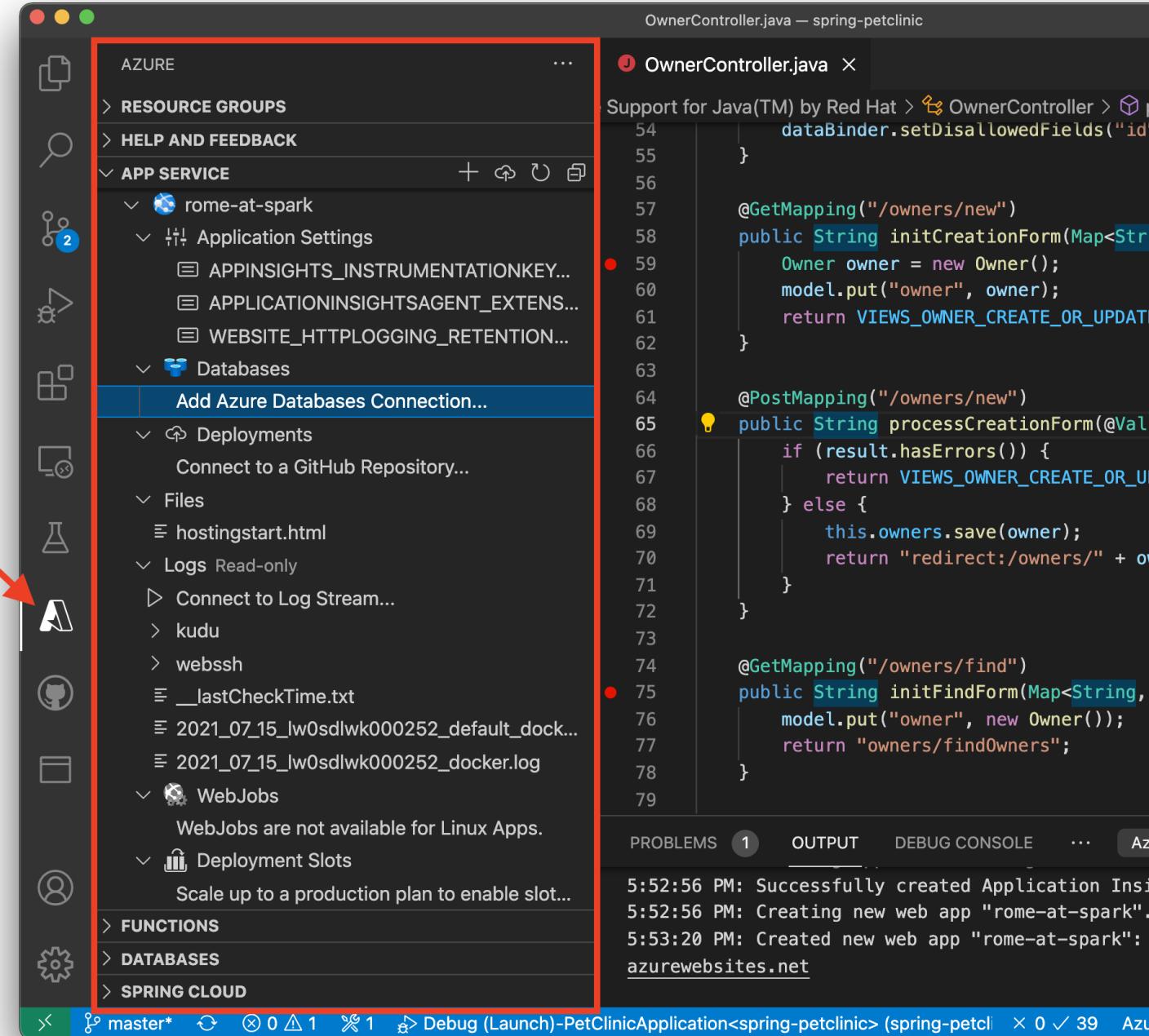
Team Collaboration

- Install **GitHub Pull Requests and Issues** from the marketplace
- **Create/Review PRs** in the **GitHub** viewlet



To the Cloud

- Use the **Azure** viewlet to manage cloud resources
- **Deploy** applications, **stream logs**, manage **DB** connections, **scale** out/up all in VS Code



The screenshot shows the VS Code interface with the Azure viewlet open. The viewlet displays a list of Azure resources for the 'rome-at-spark' app service, including Resource Groups, Help and Feedback, App Service (with 'Application Settings' expanded), Databases (with 'Add Azure Databases Connection...' selected), Deployments, Files, Logs, WebJobs, and Deployment Slots. A red arrow points to the Azure icon in the left sidebar. The main editor area shows Java code for 'OwnerController.java' related to a Spring PetClinic application. The bottom status bar indicates the file is 'Debug (Launch)-PetClinicApplication<spring-petclinic>' and shows a terminal tab with log output.

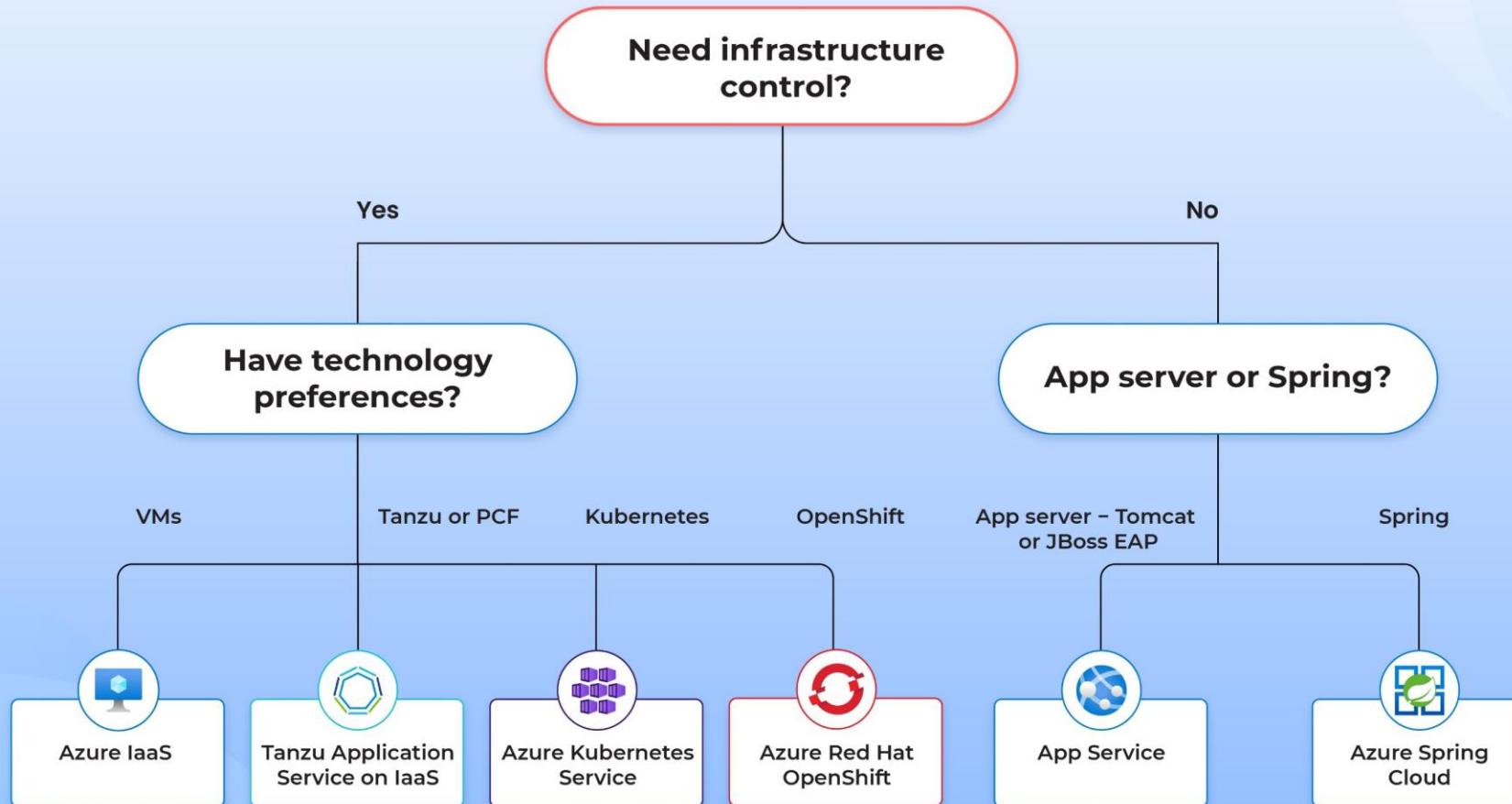
```
OwnerController.java — spring-petclinic
Support for Java(TM) by Red Hat > OwnerController > ...
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
OwnerController.java X
@GetMapping("/owners/new")
public String initCreationForm(Map<String, Object> model) {
    Owner owner = new Owner();
    model.put("owner", owner);
    return VIEWS_OWNER_CREATE_OR_UPDATE_FORM;
}

@PostMapping("/owners/new")
public String processCreationForm(@Valid @ModelAttribute("owner") Owner owner, BindingResult result) {
    if (result.hasErrors()) {
        return VIEWS_OWNER_CREATE_OR_UPDATE_FORM;
    } else {
        this.owners.save(owner);
        return "redirect:/owners/" + owner.getId();
    }
}

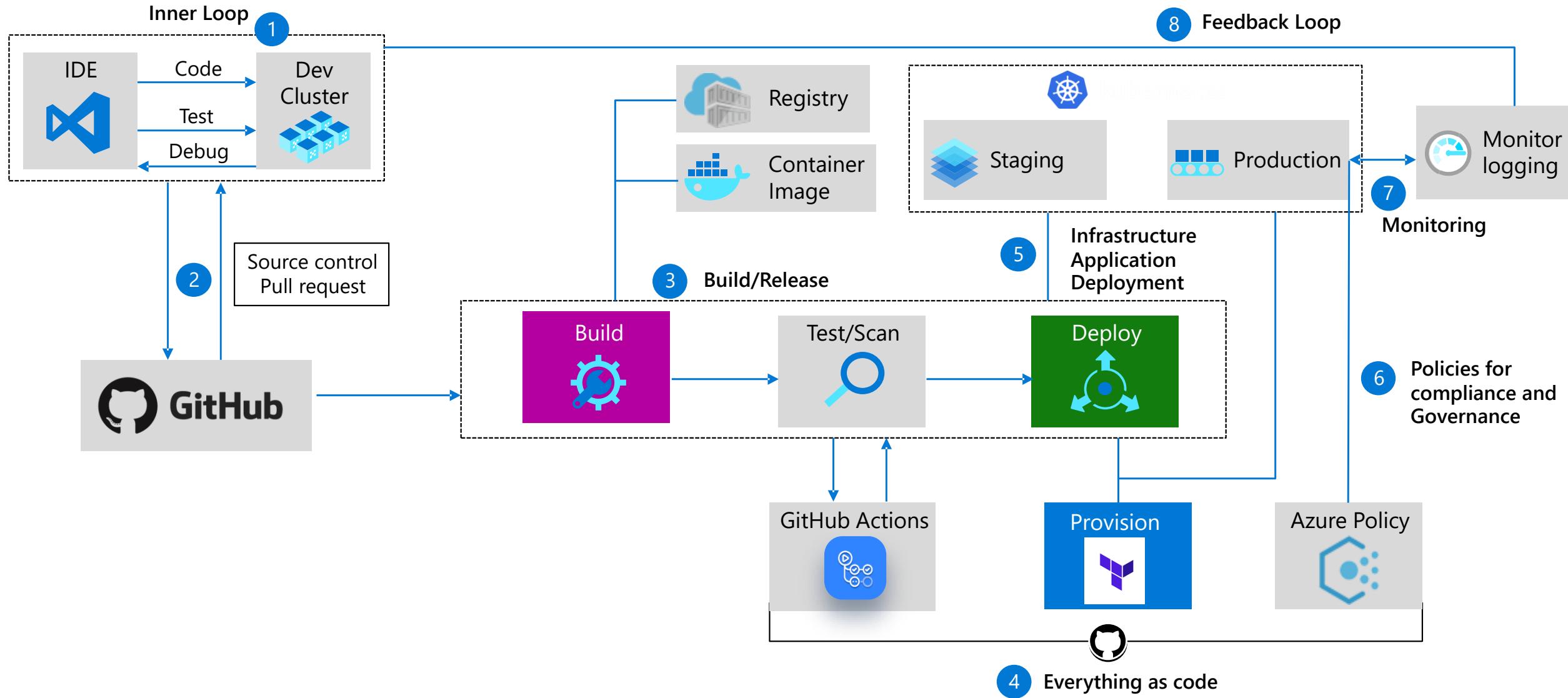
@GetMapping("/owners/find")
public String initFindForm(Map<String, Object> model) {
    model.put("owner", new Owner());
    return "owners/findOwners";
}

PROBLEMS 1 OUTPUT DEBUG CONSOLE ...
5:52:56 PM: Successfully created Application Insights resource
5:52:56 PM: Creating new web app "rome-at-spark"
5:53:20 PM: Created new web app "rome-at-spark": azurewebsites.net
master* 0 △ 1 ✘ 1 Debug (Launch)-PetClinicApplication<spring-petclinic> (spring-petclinic) 0 ✓ 39 Azur...
```

Choose an Azure Destination for Java Applications



Everything as code



Automatic scaling and distributed tracing



app-demo-201106152713 | Scale out (App Service plan)

App Service

Save Discard Refresh Logs Provide feedback

Configure Run history JSON Notify Diagnostics settings

Observed instance count - this chart plots the instance count as observed by the auto scale engine. If the chart is empty it either means auto scale is in period of time or auto scale was not configured.

Show data for last 1 hour 6 hours 12 hours 1 day 7 days Custom Pin to dashboard

Observed Capacity (Max) jbosseaprz-autoscale-628 3

Autoscale events for this time range [View more details in the Activity Log](#)

Operation name	Status	Time	Time stamp
> Autoscale scale down completed	Succeeded	10 minutes ...	Mon Nov 09 2020 12:45:02 ...
Flapping	Succeeded	11 minutes ...	Mon Nov 09 2020 12:43:59 ...
Flapping	Succeeded	12 minutes ...	Mon Nov 09 2020 12:42:59 ...
Flapping	Succeeded	12 minutes ...	Mon Nov 09 2020 12:42:15 ...
> Autoscale scale down completed	Succeeded	14 minutes ...	Mon Nov 09 2020 12:40:04 ...
> Autoscale scale up completed	Succeeded	22 minutes ...	Mon Nov 09 2020 12:32:11 ...

Home > fabrikamprod - Application map

fabrikamprod - Application map Application Insights - Last 30 days - fabrikamprod

Search (Ctrl+)

Refresh Feedback Suggest an idea Learn more Troubleshooting

NEW! Click "Filter by this node" in the right panel to filter the map to a single node and its connections. Click to dismiss.

Last 30 days

Update map components

Overview Activity log Access control (IAM) Tags Diagnose and solve problems

Investigate

Application map Smart Detection Live Metrics Stream Metrics Search Availability Failures Performance Servers Browser Troubleshooting guides (pre...) Workbooks

Usage

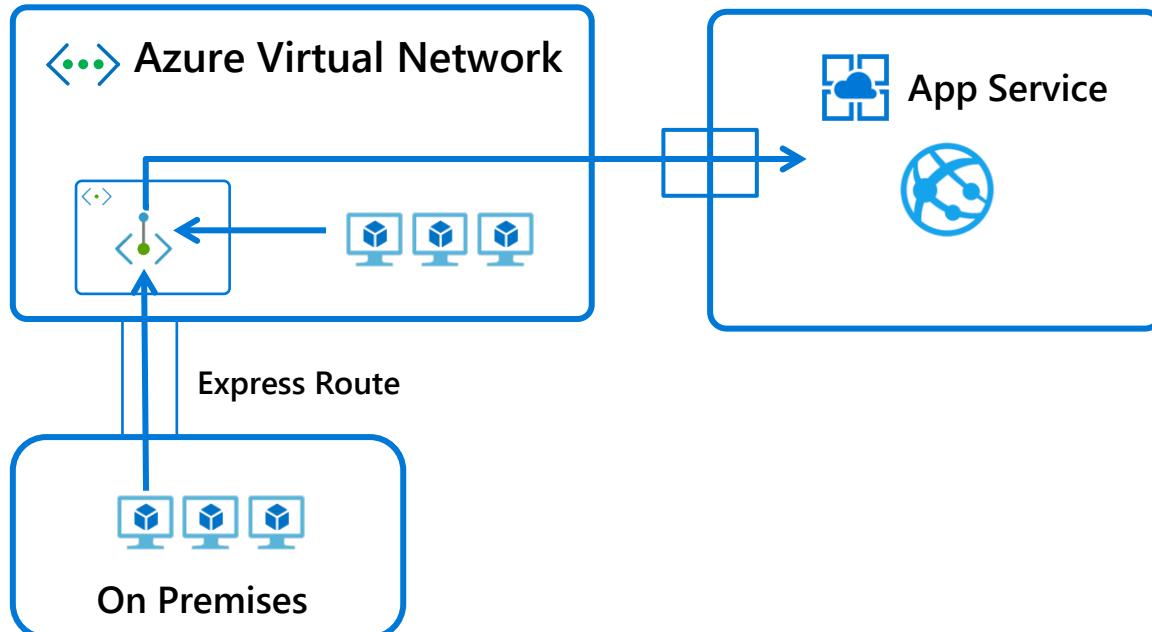
Users Sessions Events

fabrikamprod CLIENT
784K views 52 calls
128.7 ms 13K calls
11.3 ms 741K calls
419 ms 1.2K calls
7.5 s 387K tests
22 ms 5% 2.5M calls
93.9 ms 66.7% 123 calls
14.1 ms 25% 988K calls
AVAILABILITY + -

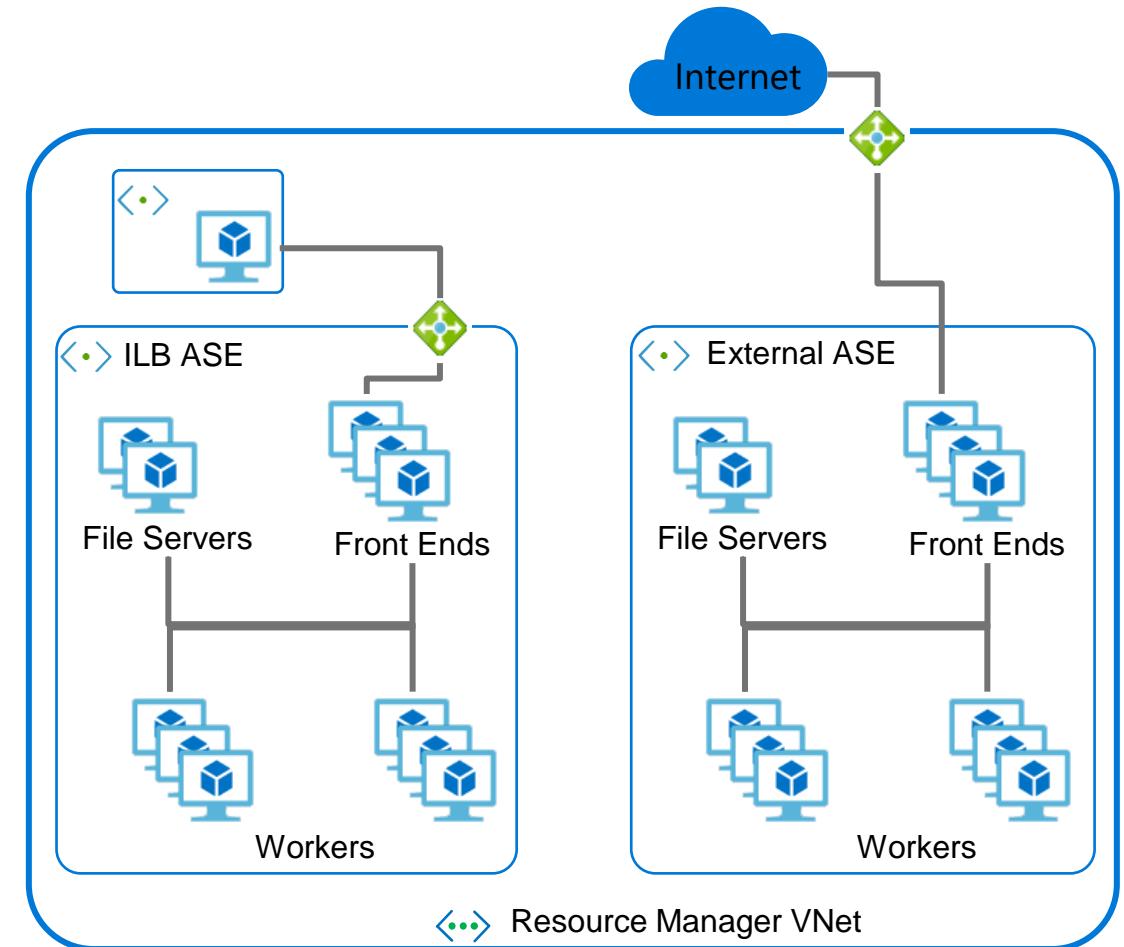
Secure your site



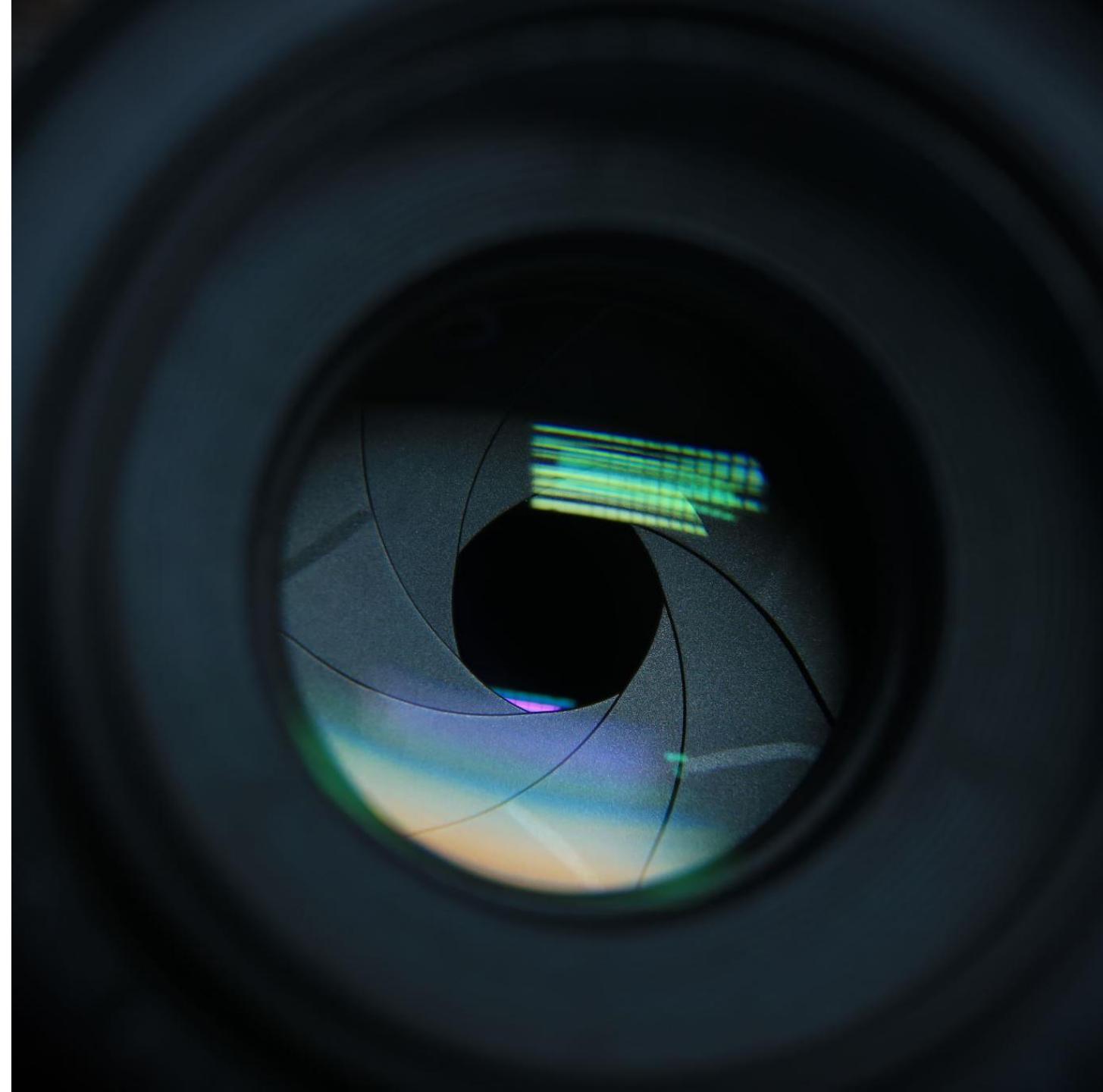
Virtual Network & Private Link



App Service Environment



Demo



2022 Roadmap

Visual Studio Code



Fundamental Inner-loop Improvement

- Improve Code Completion Smartness
 - Improve code completion intelligence and optimize for common cases
 - Better code snippet generation
 - Provide various shortcut based on user preference
- Debugging Experience Improvement
 - Debug decompiled classes
 - Faster evaluation in variable views
 - Lambda expression evaluation
 - Virtual thread support
- Testing Improvement
 - Support testing coverage
- Java 18 support

Performance & Reliability

- Improve reliability of Java Language Server
- Improve project import time
- Further improve performance for code completion

Build Tools

- Continuous Maven and Gradle improvement

Spring Boot End-to-end Improvement

- Easier creation of Spring components (including Spring project and beans)
- Boilerplate code generation for Spring classes and component
- Visualization of core Spring concepts such as beans and API endpoints
- Better Spring application lifecycle management
- Improve discoverability of Spring libraries when managing dependencies

User Experience

- Make features more discoverable and easier to use
- Make it easier for users to migrate from other Java IDEs
- Better support for students
 - Better Junit testing experience
 - Better package import / project creation experience related to Swing / JavaFX applications
 - Support Live Share in VS Code Java

Cloud-native development

- Deeper integration with Kubernetes and various cloud services

Next Steps

<https://aka.ms/java-learn-path>

<https://aka.ms/vs-code-java>

<https://aka.ms/spring-petclinic>

OR Scan Code

