the minimum number of nodes (e.g., 1–2).
2. Connect to the AKS cluster using Azure Cloud Shell with kubectl.
3. Deploy a lightweight Nginx application for verification.
4. Delete the AKS cluster immediately after testing to avoid additional VM and cluster costs.

Practical Task 6: Deploy a Containerized Application on AKS

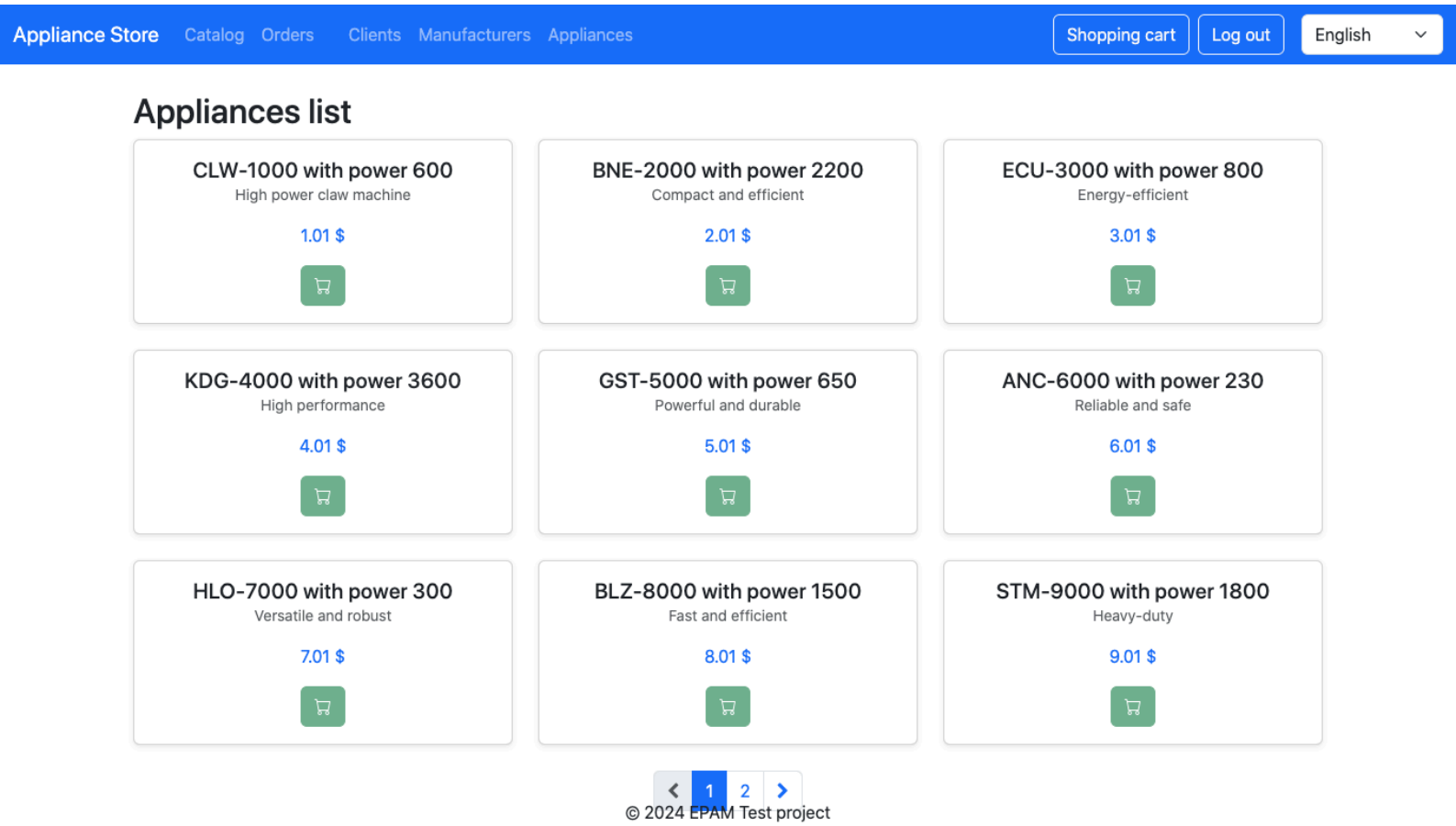
Requirements:

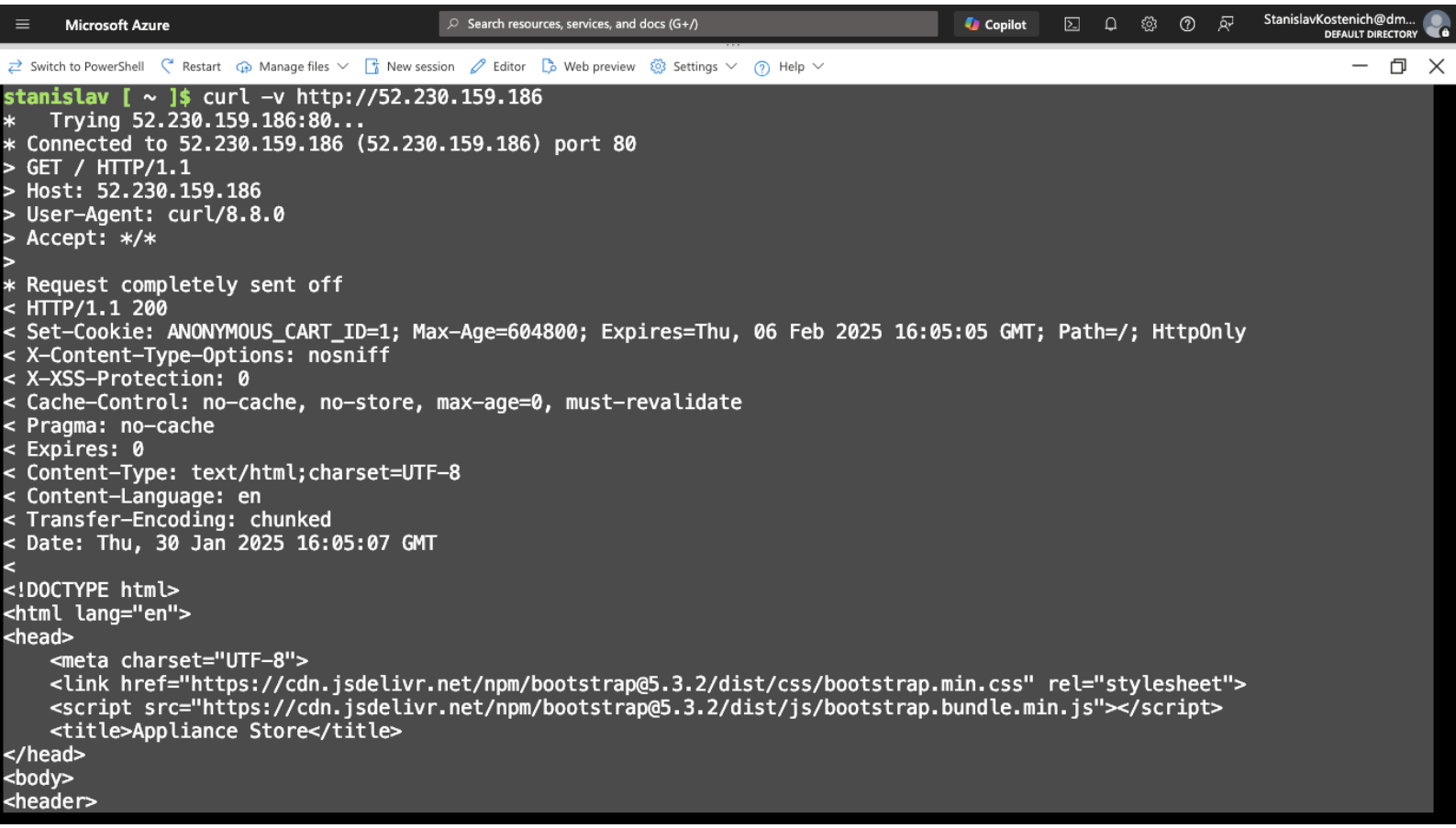
1. Build a lightweight Docker image for a simple web application (e.g., a Node.js app with minimal dependencies) and push it to Azure Container Registry (ACR).
2. Reuse the AKS cluster from Task 5 to deploy the application using a Kubernetes deployment and service manifest file.
3. Test the application for a limited time and remove the deployment afterward.

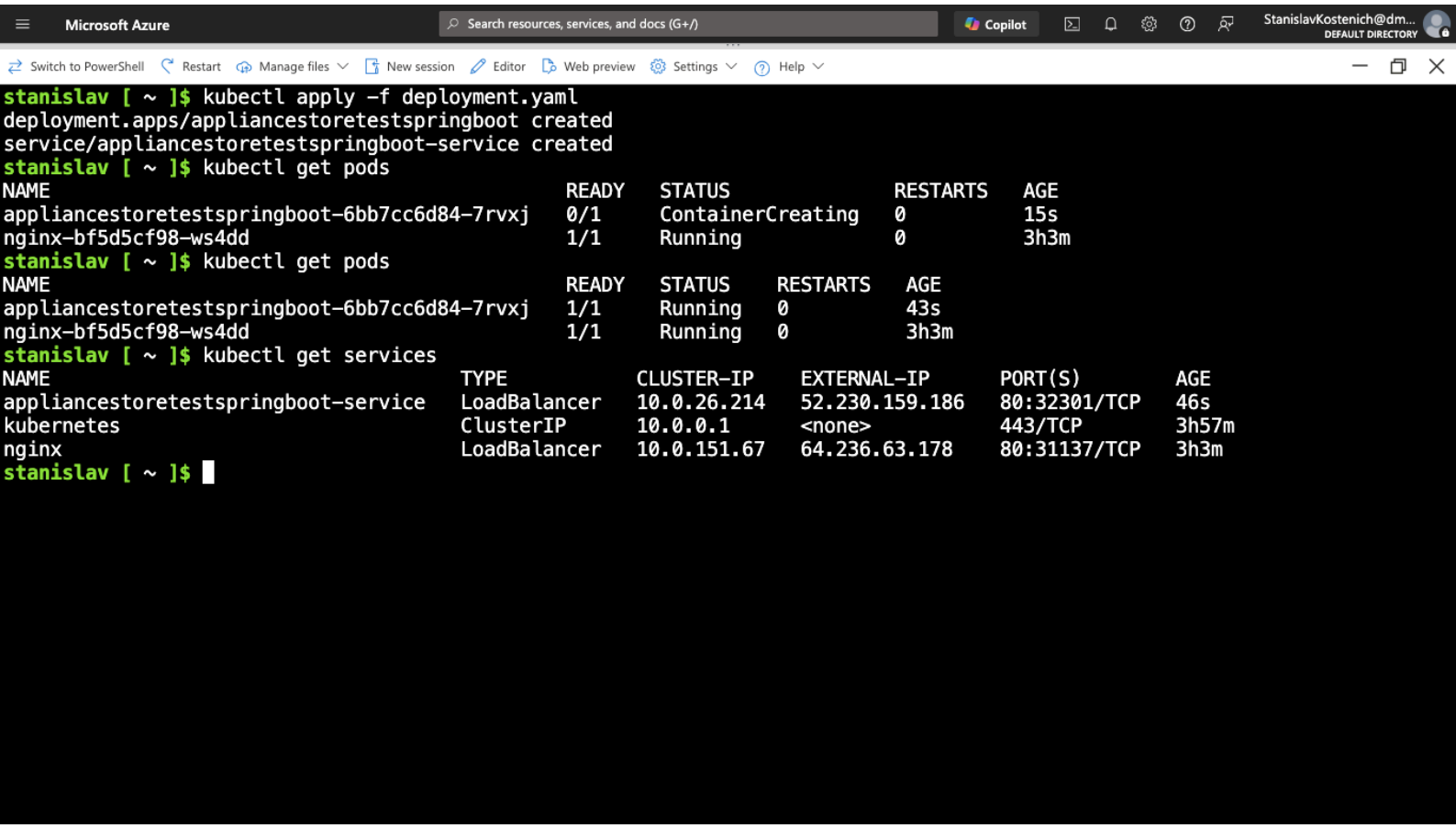
Practical Task 7: Configure and Use ConfigMaps and Secrets in AKS

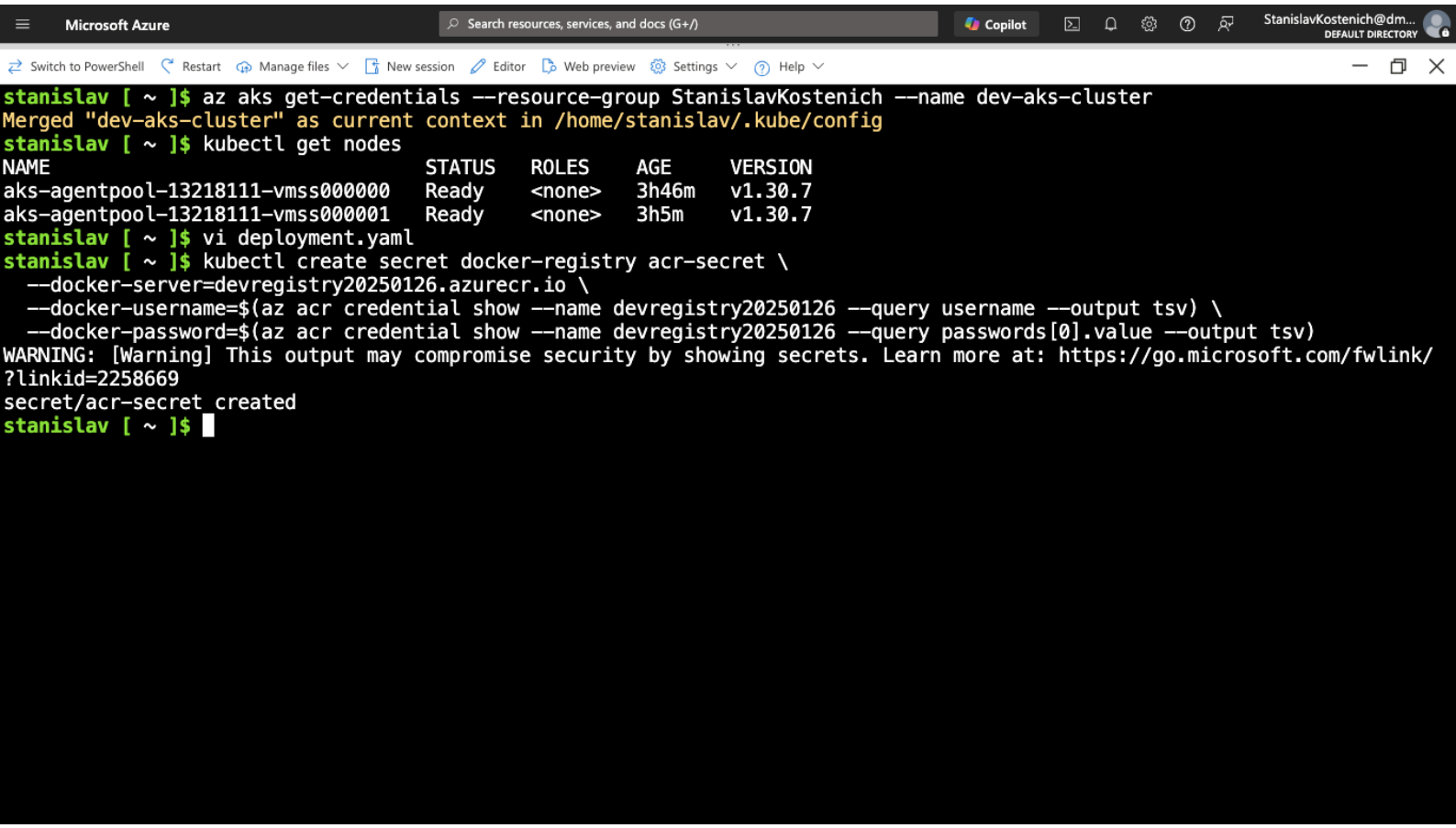
Requirements:

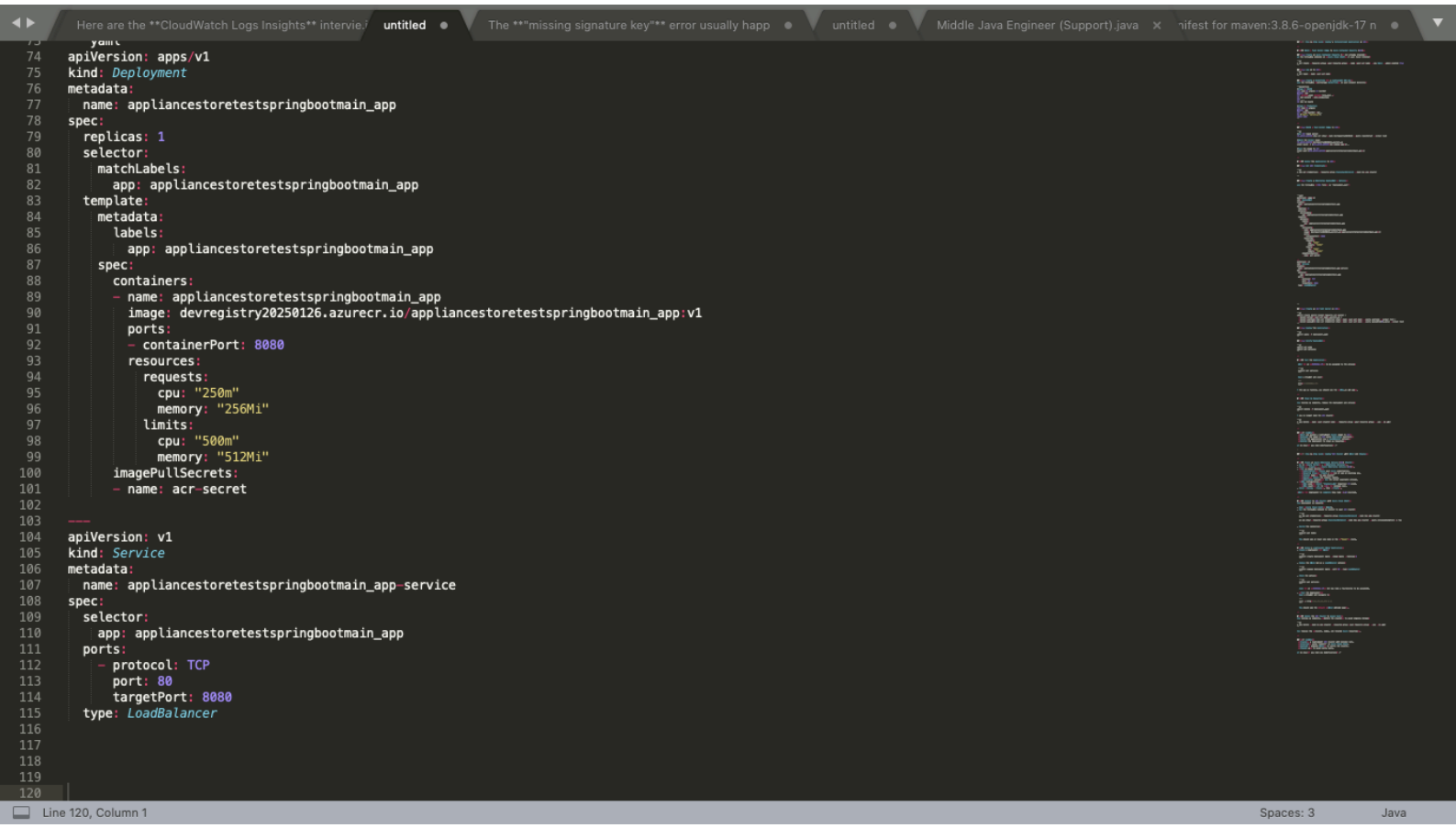
1. Create a ConfigMap to store non-sensitive configuration data with only the required key-value pairs for the application.
2. Create a Kubernetes Secret to store sensitive data (e.g., API keys) with the least amount of information needed.
3. Update the application deployment to use the ConfigMap and Secret.

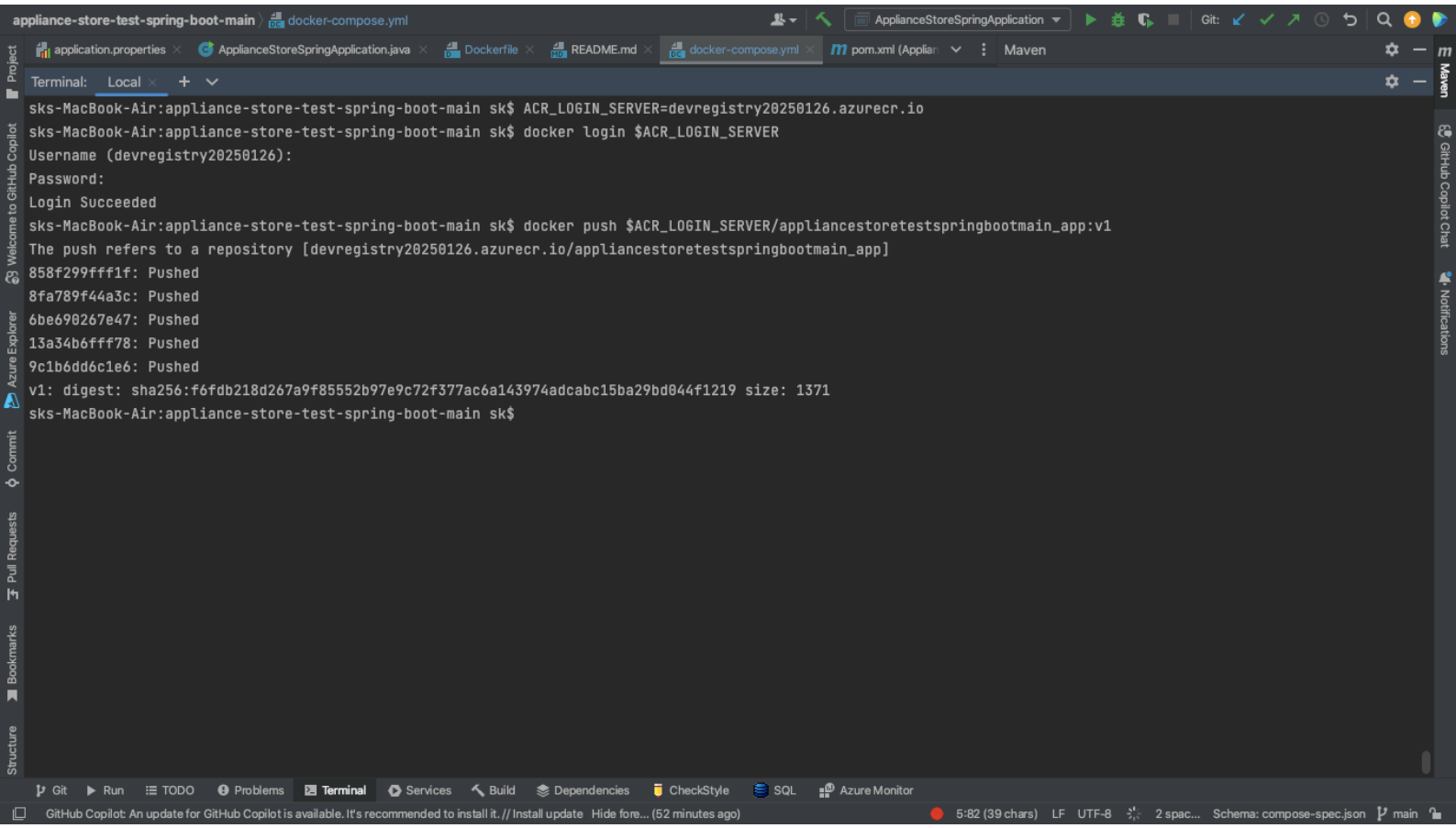

Appliance Store
Catalog
Orders
Clients
Manufacturers
Appliances
Shopping cart
Log out
English
Appliances list
CLW-1000 with power 600
High power claw machine
1.01 $
BNE-2000 with power 2200
Compact and efficient
2.01 $
ECU-3000 with power 800
Energy-efficient
3.01 $
KDG-4000 with power 3600
High performance
4.01 $
GST-5000 with power 650
Powerful and durable
5.01 $
ANC-6000 with power 230
Reliable and safe
6.01 $
HLO-7000 with power 300
Versatile and robust
7.01 $
BLZ-8000 with power 1500
Fast and efficient
8.01 $
STM-9000 with power 1800
Heavy-duty
9.01 $
1
2
© 2024 EPAM Test project

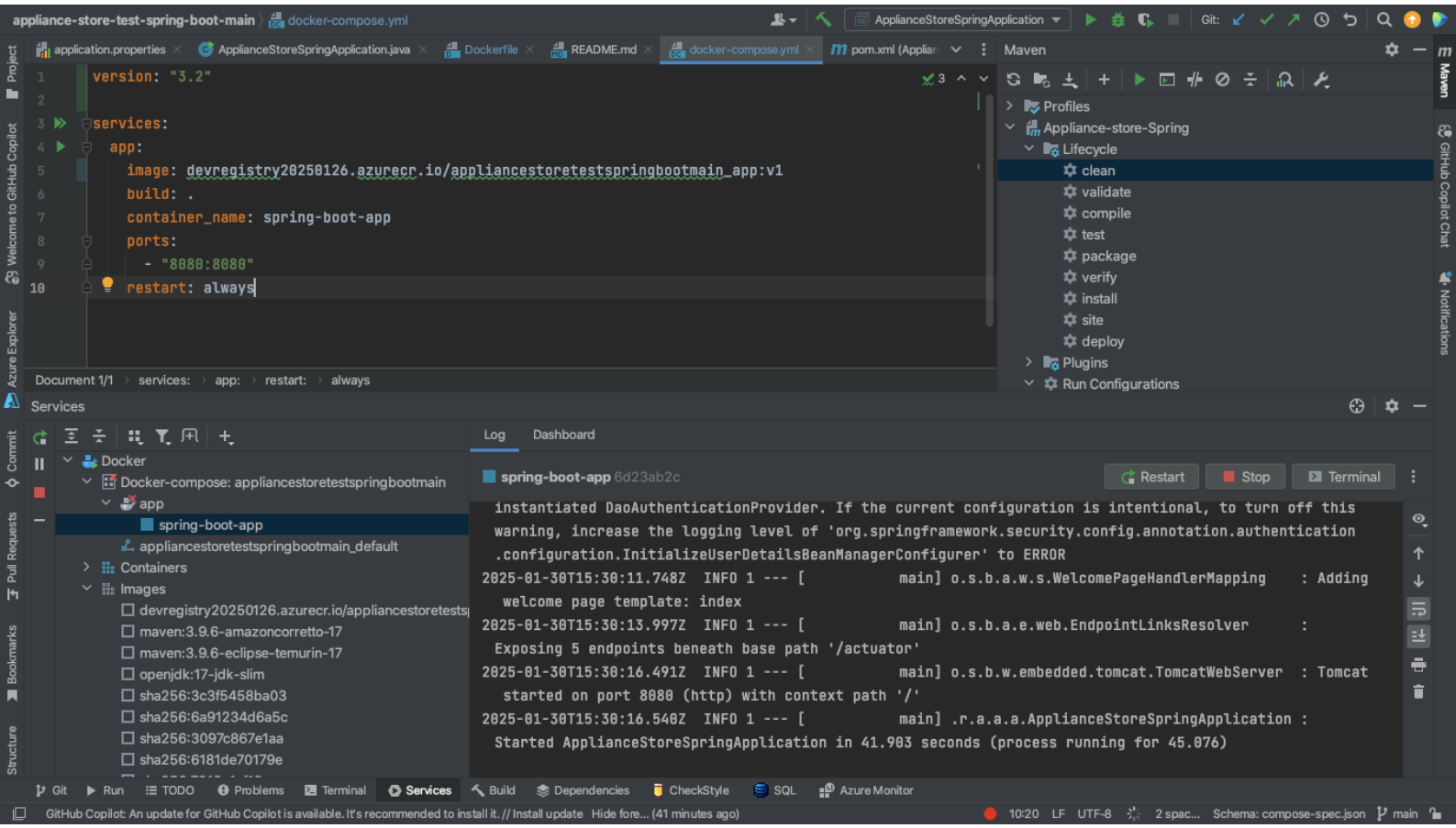

Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
StanislavKostenich@dm...
DEFAULT DIRECTORY
Switch to PowerShell
Restart
Manage files
New session
Editor
Web preview
Settings
Help
stanislav [~]$ curl -v http://52.230.159.186
*   Trying 52.230.159.186:80...
* Connected to 52.230.159.186 (52.230.159.186) port 80
> GET / HTTP/1.1
> Host: 52.230.159.186
> User-Agent: curl/8.8.0
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200
< Set-Cookie: ANONYMOUS_CART_ID=1; Max-Age=604800; Expires=Thu, 06 Feb 2025 16:05:05 GMT; Path=/; HttpOnly
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 0
< Cache-Control: no-cache, no-store, max-age=0, must-revalidate
< Pragma: no-cache
< Expires: 0
< Content-Type: text/html;charset=UTF-8
< Content-Language: en
< Transfer-Encoding: chunked
< Date: Thu, 30 Jan 2025 16:05:07 GMT
<
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
    <title>Appliance Store</title>
</head>
<body>
<header>

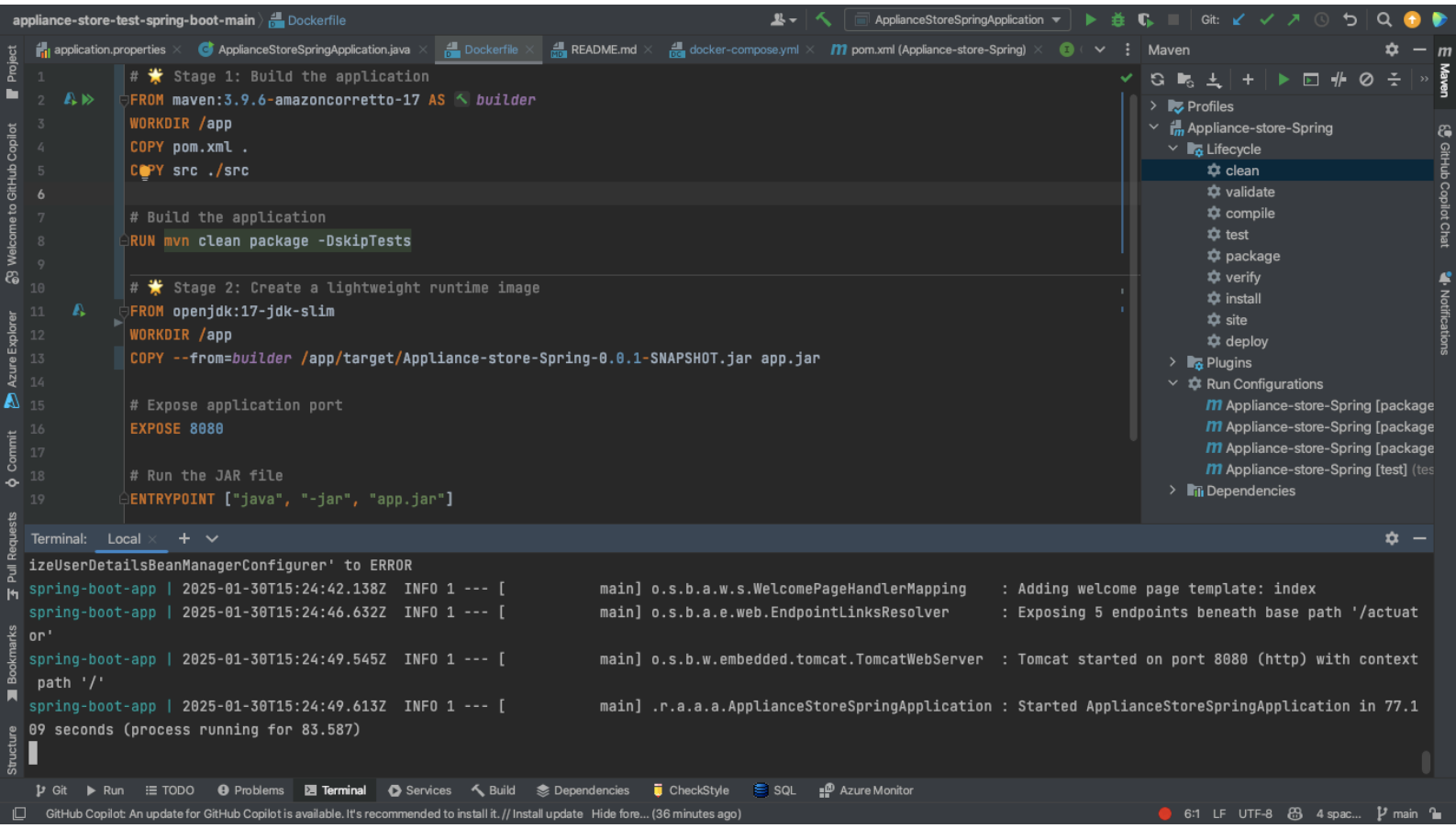

Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
StanislavKostenich@dm...
DEFAULT DIRECTORY
Switch to PowerShell
Restart
Manage files
New session
Editor
Web preview
Settings
Help
stanislav [~]$ kubectl apply -f deployment.yaml
deployment.apps/appliancestoretestspringboot created
service/appliancestoretestspringboot-service created
stanislav [~]$ kubectl get pods
NAME READY STATUS RESTARTS AGE
appliancestoretestspringboot-6bb7cc6d84-7rvxj 0/1 ContainerCreating 0 15s
nginx-bf5d5cf98-ws4dd 1/1 Running 0 3h3m
stanislav [~]$ kubectl get pods
NAME READY STATUS RESTARTS AGE
appliancestoretestspringboot-6bb7cc6d84-7rvxj 1/1 Running 0 43s
nginx-bf5d5cf98-ws4dd 1/1 Running 0 3h3m
stanislav [~]$ kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
appliancestoretestspringboot-service LoadBalancer 10.0.26.214 52.230.159.186 80:32301/TCP 46s
kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 3h57m
nginx LoadBalancer 10.0.151.67 64.236.63.178 80:31137/TCP 3h3m
stanislav [~]$


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
StanislavKostenich@dm...
DEFAULT DIRECTORY
Switch to PowerShell
Restart
Manage files
New session
Editor
Web preview
Settings
Help
stanislav [~]$ az aks get-credentials --resource-group StanislavKostenich --name dev-aks-cluster
Merged "dev-aks-cluster" as current context in /home/stanislav/.kube/config
stanislav [~]$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
aks-agentpool-13218111-vmss000000 Ready <none> 3h46m v1.30.7
aks-agentpool-13218111-vmss000001 Ready <none> 3h5m v1.30.7
stanislav [~]$ vi deployment.yaml
stanislav [~]$ kubectl create secret docker-registry acr-secret \
--docker-server=devregistry20250126.azurecr.io \
--docker-username=$(az acr credential show --name devregistry20250126 --query username --output tsv) \
--docker-password=$(az acr credential show --name devregistry20250126 --query passwords[0].value --output tsv)
WARNING: [Warning] This output may compromise security by showing secrets. Learn more at: https://go.microsoft.com/fwlink/
?linkid=2258669
secret/acr-secret created
stanislav [~]$


Here are the **CloudWatch Logs Insights** intervie
untitled
The ***missing signature key*** error usually happ
untitled
Middle Java Engineer (Support).java
nifest for maven:3.8.6-openjdk-17 n
yaml
apiVersion: apps/v1
kind: Deployment
metadata:
name: appliancestoretestspringbootmain_app
spec:
replicas: 1
selector:
matchLabels:
app: appliancestoretestspringbootmain_app
template:
metadata:
labels:
app: appliancestoretestspringbootmain_app
spec:
containers:
- name: appliancestoretestspringbootmain_app
image: devregistry20250126.azurecr.io/appliancestoretestspringbootmain_app:v1
ports:
- containerPort: 8080
resources:
requests:
cpu: "250m"
memory: "256Mi"
limits:
cpu: "500m"
memory: "512Mi"
imagePullSecrets:
- name: acr-secret
---
apiVersion: v1
kind: Service
metadata:
name: appliancestoretestspringbootmain_app-service
spec:
selector:
app: appliancestoretestspringbootmain_app
ports:
- protocol: TCP
port: 80
targetPort: 8080
type: LoadBalancer
Line 120, Column 1
Spaces: 3
Java


appliance-store-test-spring-boot-main
docker-compose.yml
ApplianceStoreSpringApplication
Git:
application.properties
ApplianceStoreSpringApplication.java
Dockerfile
README.md
docker-compose.yml
pom.xml (Applian
Maven
Terminal:
Local
sks-MacBook-Air:appliance-store-test-spring-boot-main sk$ ACR_LOGIN_SERVER=devregistry20250126.azurecr.io
sks-MacBook-Air:appliance-store-test-spring-boot-main sk$ docker login $ACR_LOGIN_SERVER
Username (devregistry20250126):
Password:
Login Succeeded
sks-MacBook-Air:appliance-store-test-spring-boot-main sk$ docker push $ACR_LOGIN_SERVER/appliancestoretestspringbootmain_app:v1
The push refers to a repository [devregistry20250126.azurecr.io/appliancestoretestspringbootmain_app]
858f299fff1f: Pushed
8fa789f44a3c: Pushed
6be690267e47: Pushed
13a34b6fff78: Pushed
9c1b6dd6c1e6: Pushed
v1: digest: sha256:f6fdb218d267a9f85552b97e9c72f377ac6a143974adcabc15ba29bd044f1219 size: 1371
sks-MacBook-Air:appliance-store-test-spring-boot-main sk$
Project
Welcome to GitHub Copilot
Azure Explorer
Commit
Pull Requests
Bookmarks
Structure
Maven
GitHub Copilot Chat
Notifications
Git
Run
TODO
Problems
Terminal
Services
Build
Dependencies
CheckStyle
SQL
Azure Monitor
GitHub Copilot: An update for GitHub Copilot is available. It's recommended to install it. // Install update
Hide fore... (52 minutes ago)
5:82 (39 chars)
LF
UTF-8
2 spac...
Schema: compose-spec.json
main


appliance-store-test-spring-boot-main
docker-compose.yml
ApplianceStoreSpringApplication
application.properties
ApplianceStoreSpringApplication.java
Dockerfile
README.md
docker-compose.yml
pom.xml (Appliar
version: "3.2"
services:
app:
image: devregistry20250126.azurecr.io/appliancestoretestspringbootmain_app:v1
build: .
container_name: spring-boot-app
ports:
- "8080:8080"
restart: always
Document 1/1 › services: › app: › restart: › always
Maven
Profiles
Appliance-store-Spring
Lifecycle
clean
validate
compile
test
package
verify
install
site
deploy
Plugins
Run Configurations
Services
Docker
Docker-compose: appliancestoretestspringbootmain
app
spring-boot-app
appliancestoretestspringbootmain_default
Containers
Images
devregistry20250126.azurecr.io/appliancestoretests
maven:3.9.6-amazoncorretto-17
maven:3.9.6-eclipse-temurin-17
openjdk:17-jdk-slim
sha256:3c3f5458ba03
sha256:6a91234d6a5c
sha256:3097c867e1aa
sha256:6181de70179e
Log
Dashboard
spring-boot-app 6d23ab2c
Restart
Stop
Terminal
instantiated DaoAuthenticationProvider. If the current configuration is intentional, to turn off this warning, increase the logging level of 'org.springframework.security.config.annotation.authentication.configuration.InitializeUserDetailsBeanManagerConfigurer' to ERROR
2025-01-30T15:30:11.748Z INFO 1 --- [main] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page template: index
2025-01-30T15:30:13.997Z INFO 1 --- [main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 5 endpoints beneath base path '/actuator'
2025-01-30T15:30:16.491Z INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'
2025-01-30T15:30:16.540Z INFO 1 --- [main] .r.a.a.a.ApplianceStoreSpringApplication : Started ApplianceStoreSpringApplication in 41.903 seconds (process running for 45.876)
Git
Run
TODO
Problems
Terminal
Services
Build
Dependencies
CheckStyle
SQL
Azure Monitor
GitHub Copilot: An update for GitHub Copilot is available. It's recommended to install it. // Install update Hide fore... (41 minutes ago)
10:20
LF
UTF-8
2 spac...
Schema: compose-spec.json
main


appliance-store-test-spring-boot-main
Dockerfile
ApplianceStoreSpringApplication
application.properties
ApplianceStoreSpringApplication.java
Dockerfile
README.md
docker-compose.yml
pom.xml (Appliance-store-Spring)
Stage 1: Build the application
FROM maven:3.9.6-amazoncorretto-17 AS builder
WORKDIR /app
COPY pom.xml .
COPY src ./src
Build the application
RUN mvn clean package -DskipTests
Stage 2: Create a lightweight runtime image
FROM openjdk:17-jdk-slim
WORKDIR /app
COPY --from=builder /app/target/Appliance-store-Spring-0.0.1-SNAPSHOT.jar app.jar
Expose application port
EXPOSE 8080
Run the JAR file
ENTRYPOINT ["java", "-jar", "app.jar"]
Maven
Profiles
Appliance-store-Spring
Lifecycle
clean
validate
compile
test
package
verify
install
site
deploy
Plugins
Run Configurations
Appliance-store-Spring [package
Appliance-store-Spring [package
Appliance-store-Spring [package
Appliance-store-Spring [test]
Dependencies
Terminal:
Local
izeUserDetailsBeanManagerConfigurer' to ERROR
spring-boot-app | 2025-01-30T15:24:42.138Z INFO 1 --- [main] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page template: index
spring-boot-app | 2025-01-30T15:24:46.632Z INFO 1 --- [main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 5 endpoints beneath base path '/actuat
or'
spring-boot-app | 2025-01-30T15:24:49.545Z INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context
path '/'
spring-boot-app | 2025-01-30T15:24:49.613Z INFO 1 --- [main] .r.a.a.a.ApplianceStoreSpringApplication : Started ApplianceStoreSpringApplication in 77.1
09 seconds (process running for 83.587)
Git
Run
TODO
Problems
Terminal
Services
Build
Dependencies
CheckStyle
SQL
Azure Monitor
GitHub Copilot: An update for GitHub Copilot is available. It's recommended to install it. // Install update Hide fore... (36 minutes ago)
6:1
LF
UTF-8
4 spac...
main

4. Delete the AKS cluster immediately after testing to avoid additional VM and cluster costs.

Practical Task 6: Deploy a Containerized Application on AKS

Requirements:

1. Build a lightweight Docker image for a simple web application (e.g., a Node.js app with minimal dependencies) and push it to Azure Container Registry (ACR).
2. Reuse the AKS cluster from Task 5 to deploy the application using a Kubernetes deployment and service manifest file.
3. Test the application for a limited time and remove the deployment afterward.

Practical Task 7: Configure and Use ConfigMaps and Secrets in AKS

Requirements:

1. Create a ConfigMap to store non-sensitive configuration data with only the required key-value pairs for the application.
2. Create a Kubernetes Secret to store sensitive data (e.g., API keys) with the least amount of