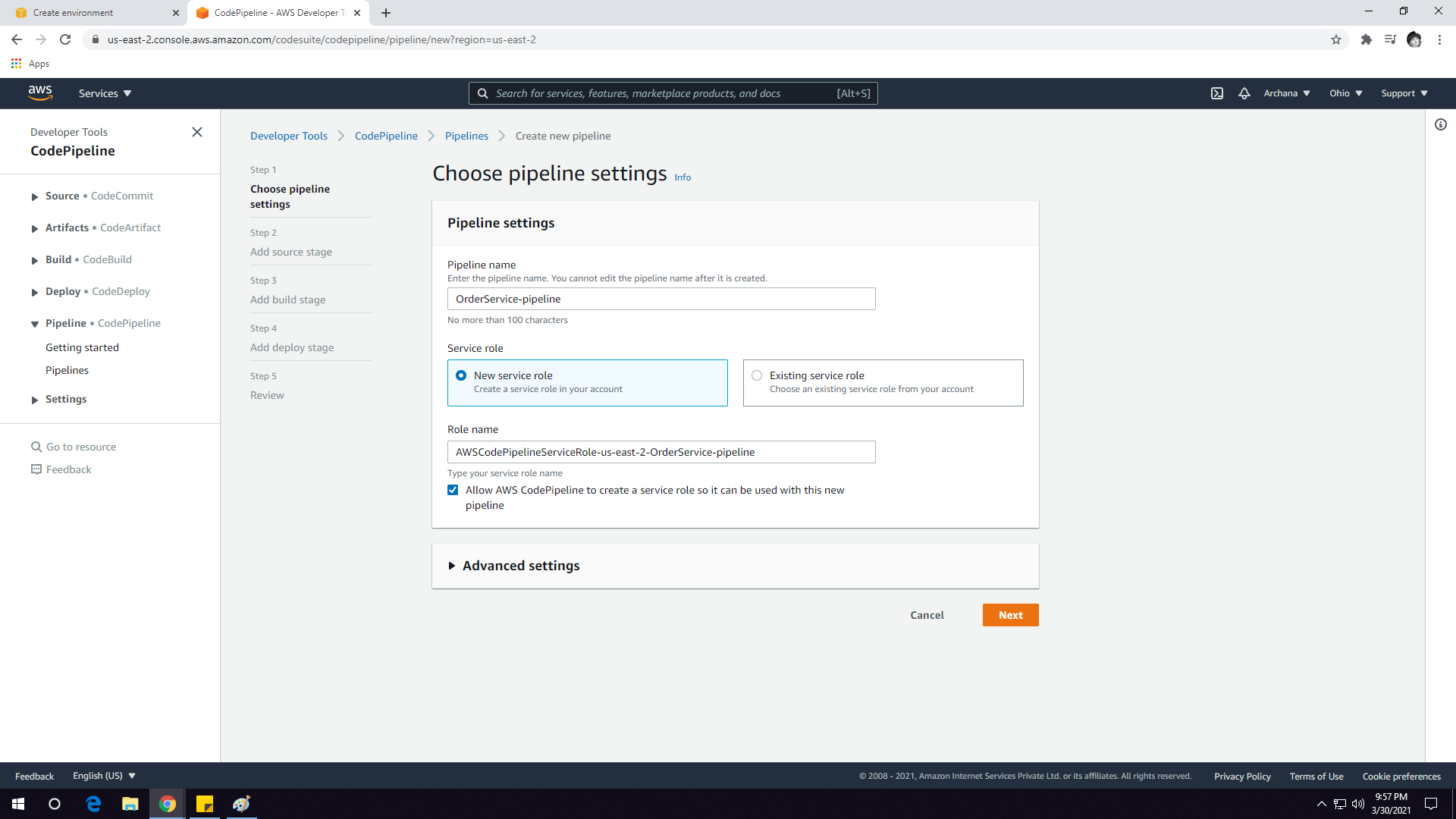
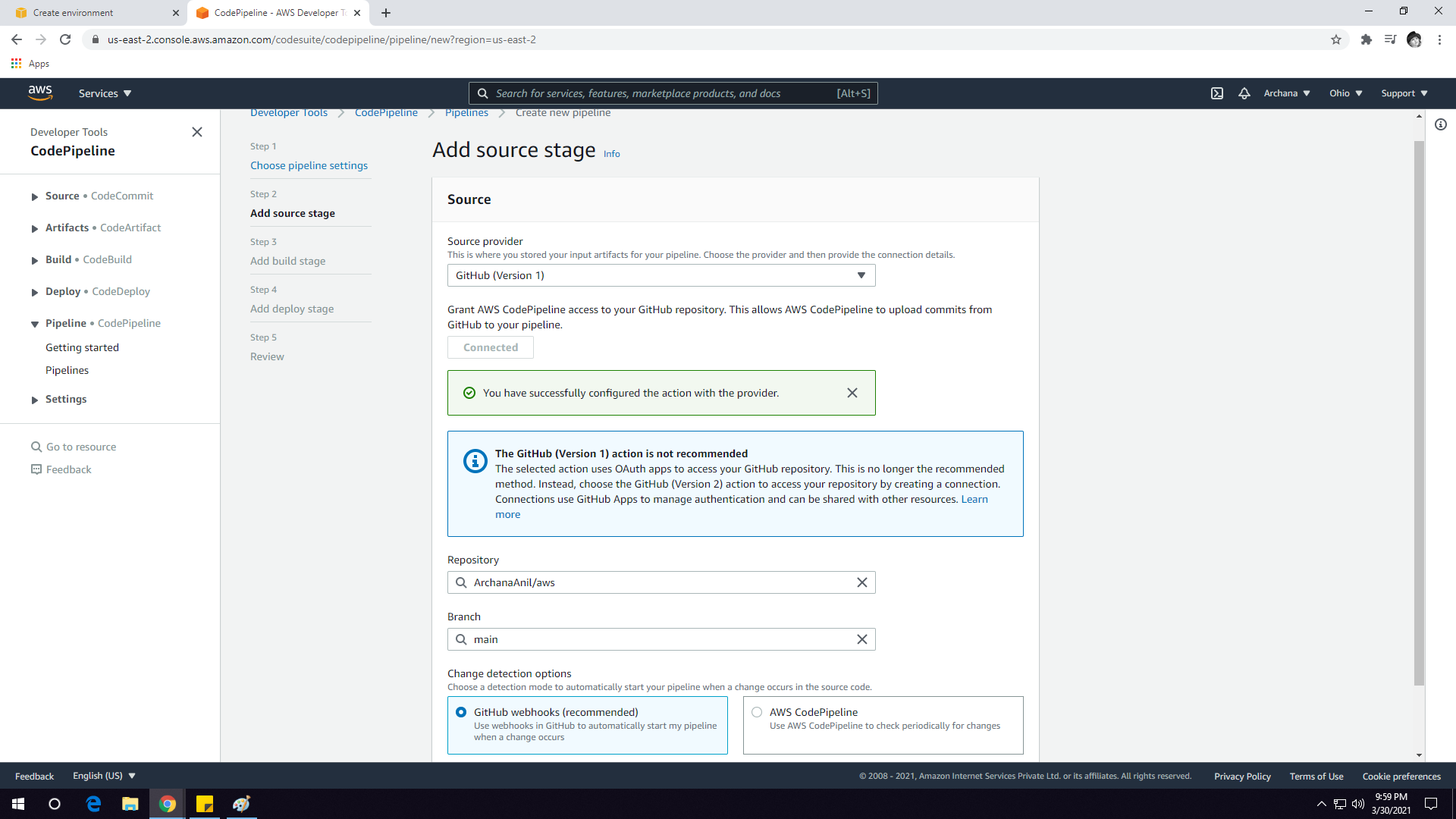
AWS – CI/CD

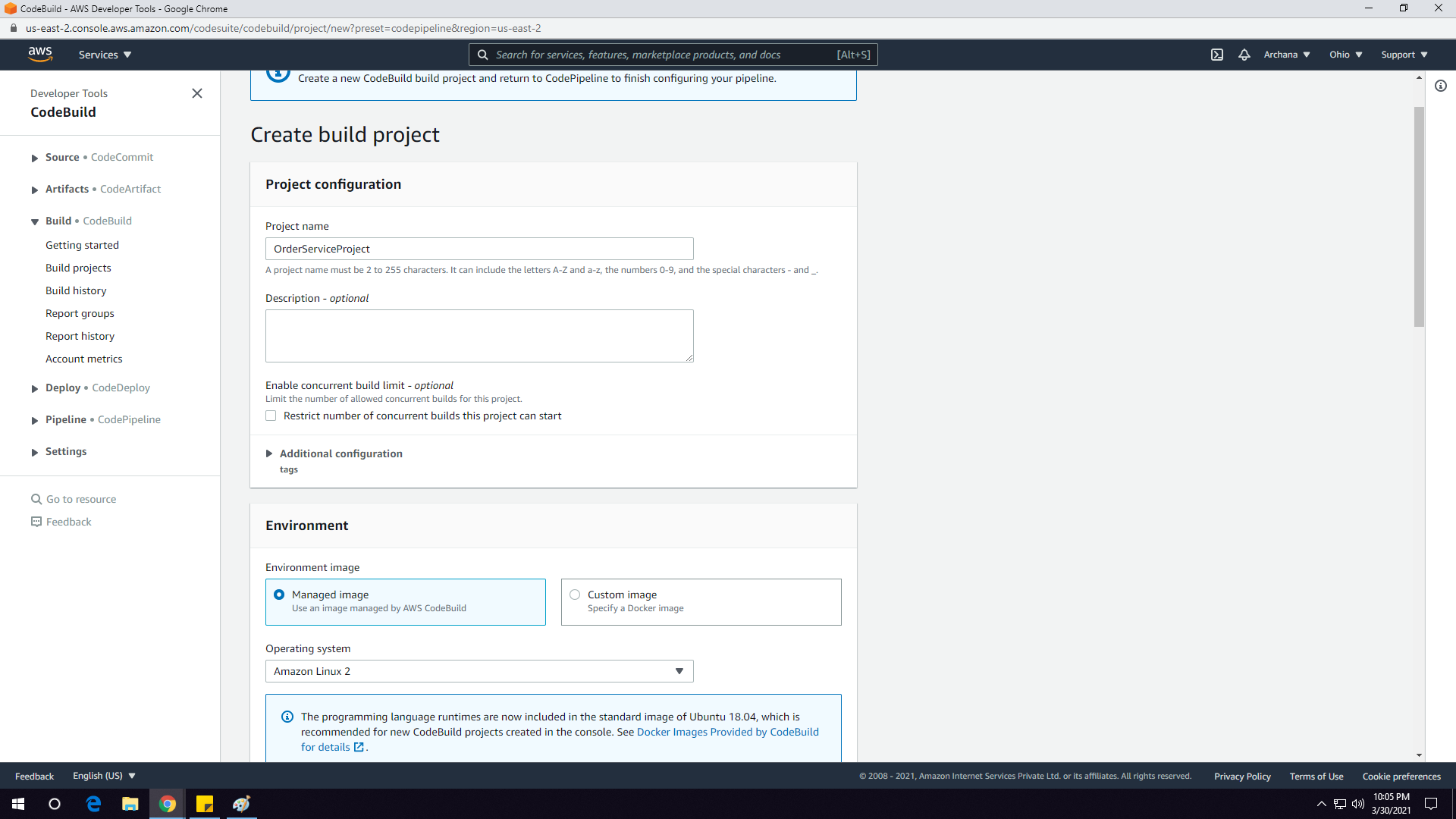
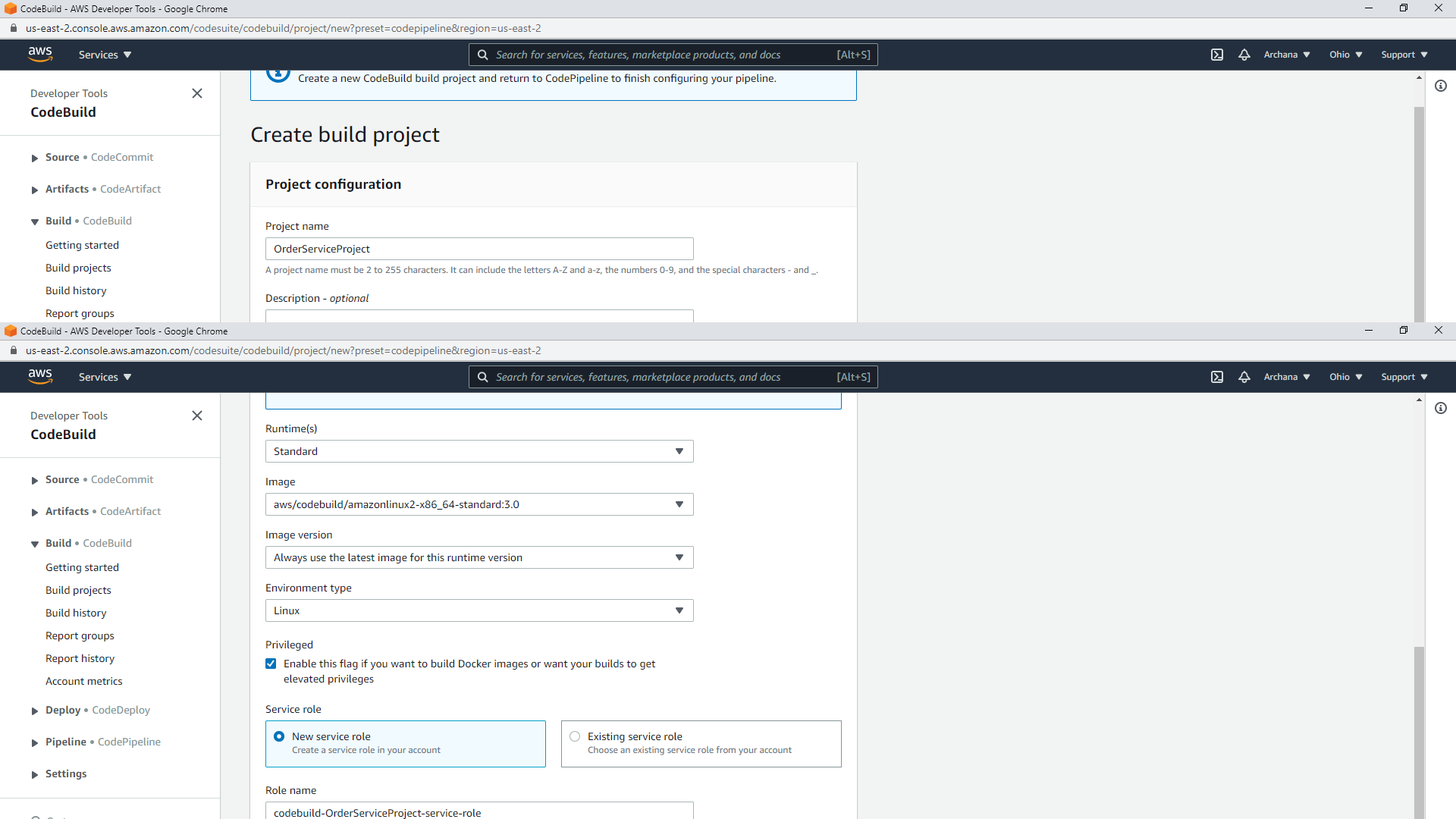
* Springboot Project

**Order.java**

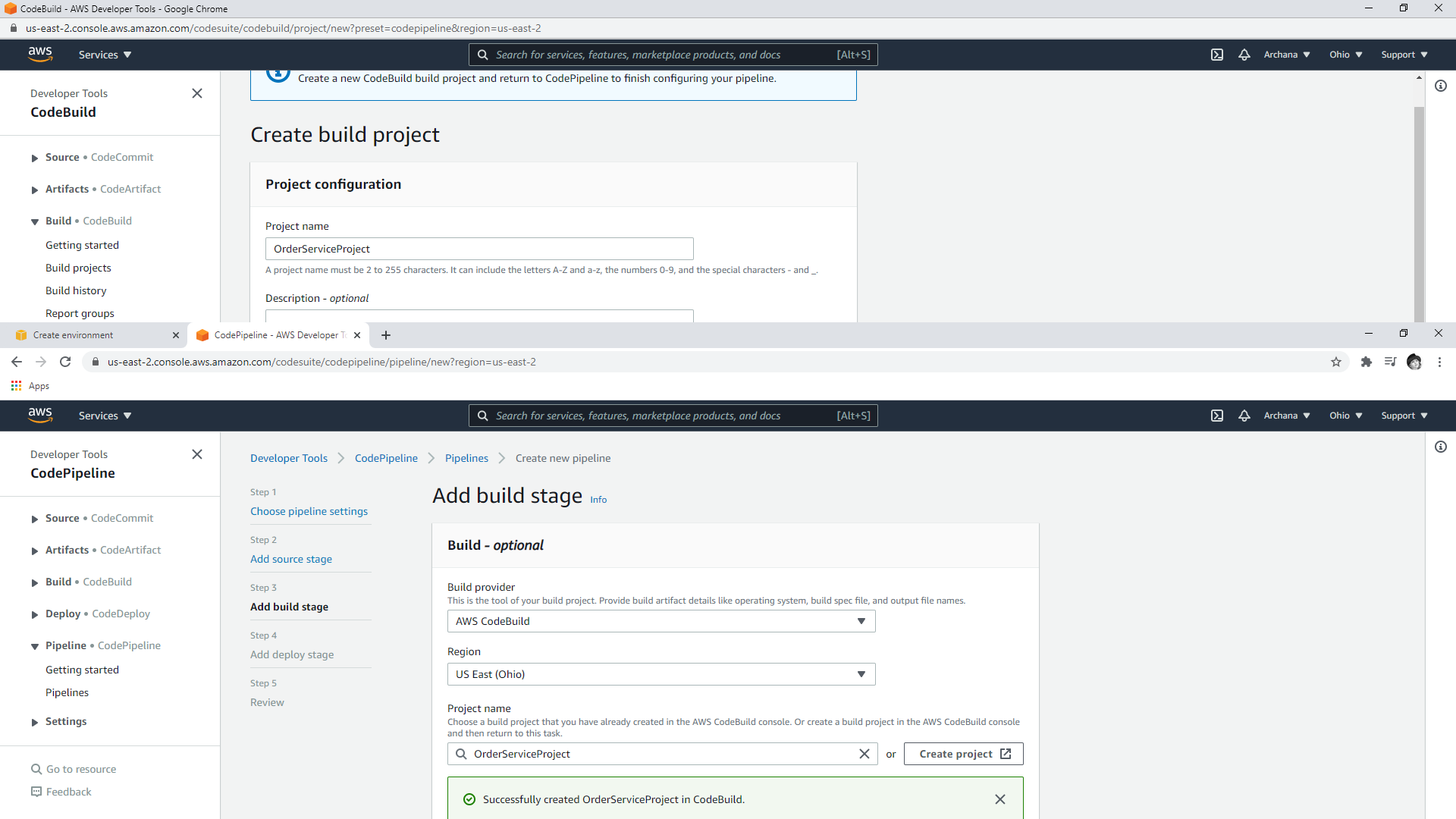
|  |
| --- |
| package com.aws;  import lombok.AllArgsConstructor;  import lombok.Data;  import lombok.NoArgsConstructor;  import lombok.ToString;  @Data  @AllArgsConstructor  @NoArgsConstructor  @ToString  public class Order {  private int id;  private String name;  private int quantity;  private long price;  }  **OrderDao.java**  package com.aws;  import org.springframework.stereotype.Repository;  import java.util.List;  import java.util.stream.Collectors;  import java.util.stream.Stream;  @Repository  public class OrderDao {  public List<Order> getOrders() {  return Stream.of(  new Order(101, "Mobile", 1, 30000),  new Order(58, "Book", 4, 2000),  new Order(205, "Laptop", 1, 150000),  new Order(809, "headset", 1, 1799))  .collect(Collectors.toList());  }  }  **OrderServiceApplication.java**  package com.aws;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.boot.SpringApplication;  import org.springframework.boot.autoconfigure.SpringBootApplication;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.RequestMapping;  import org.springframework.web.bind.annotation.RestController;  import java.util.Comparator;  import java.util.List;  import java.util.stream.Collectors;  @SpringBootApplication  @RestController  @RequestMapping("/orders")  public class OrderServiceApplication {  @Autowired  private OrderDao orderDao;  @GetMapping  public List<Order> fetchOrders() {  return orderDao.getOrders().stream().  sorted(Comparator.comparing(Order::getPrice)).collect(Collectors.toList());  }  public static void main(String[] args) {  SpringApplication.run(OrderServiceApplication.class, args);  }  }  **buildspec.yml** |
|  |  |
| version: 0.2  phases:  build:  commands:  - echo Entered the build phase...  - mvn clean package  post\_build:  commands:  - echo Build completed successfully  artifacts:  files:  - target/order-service.jar  discard-paths: yes |  |
|  |  |
|  |  |
|  |  |
|  |  |
| * Creating Elastic beanstalk environment   C:\Users\888623\Desktop\1.png |  |
|  |  |
|  |  |
| C:\Users\888623\Desktop\2.png |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* Creating AWS CodePipeline

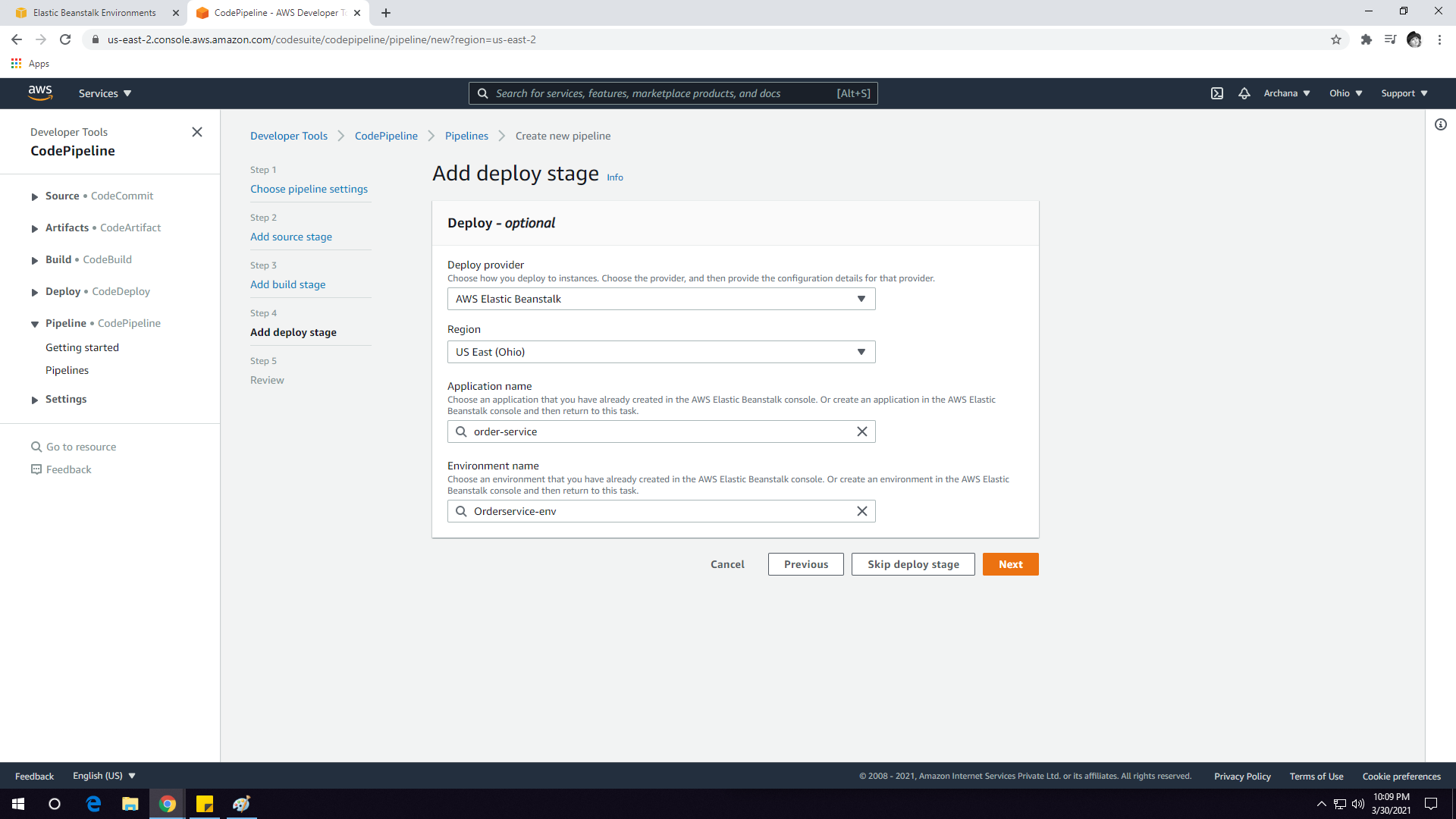




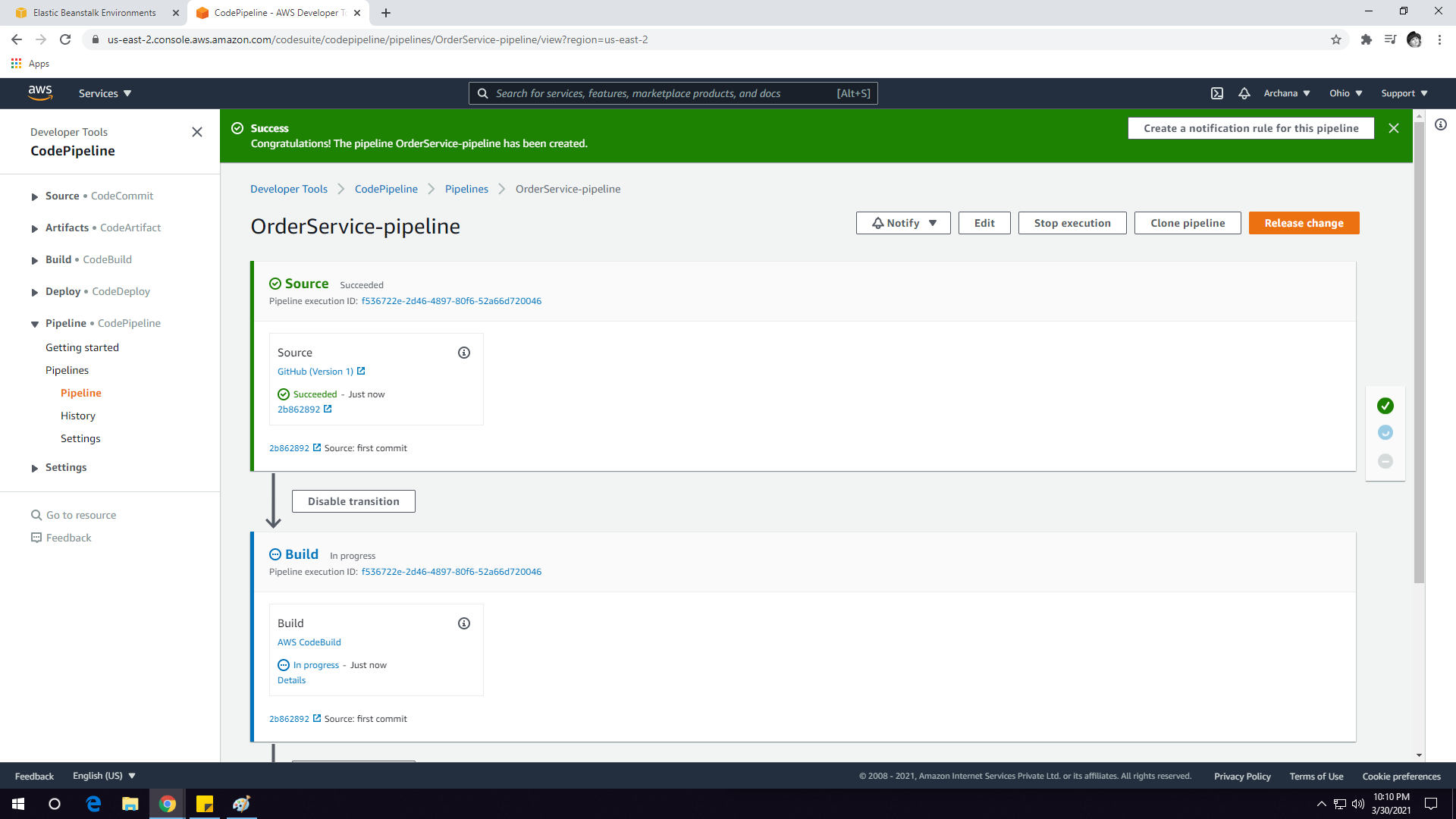
Adding the build project



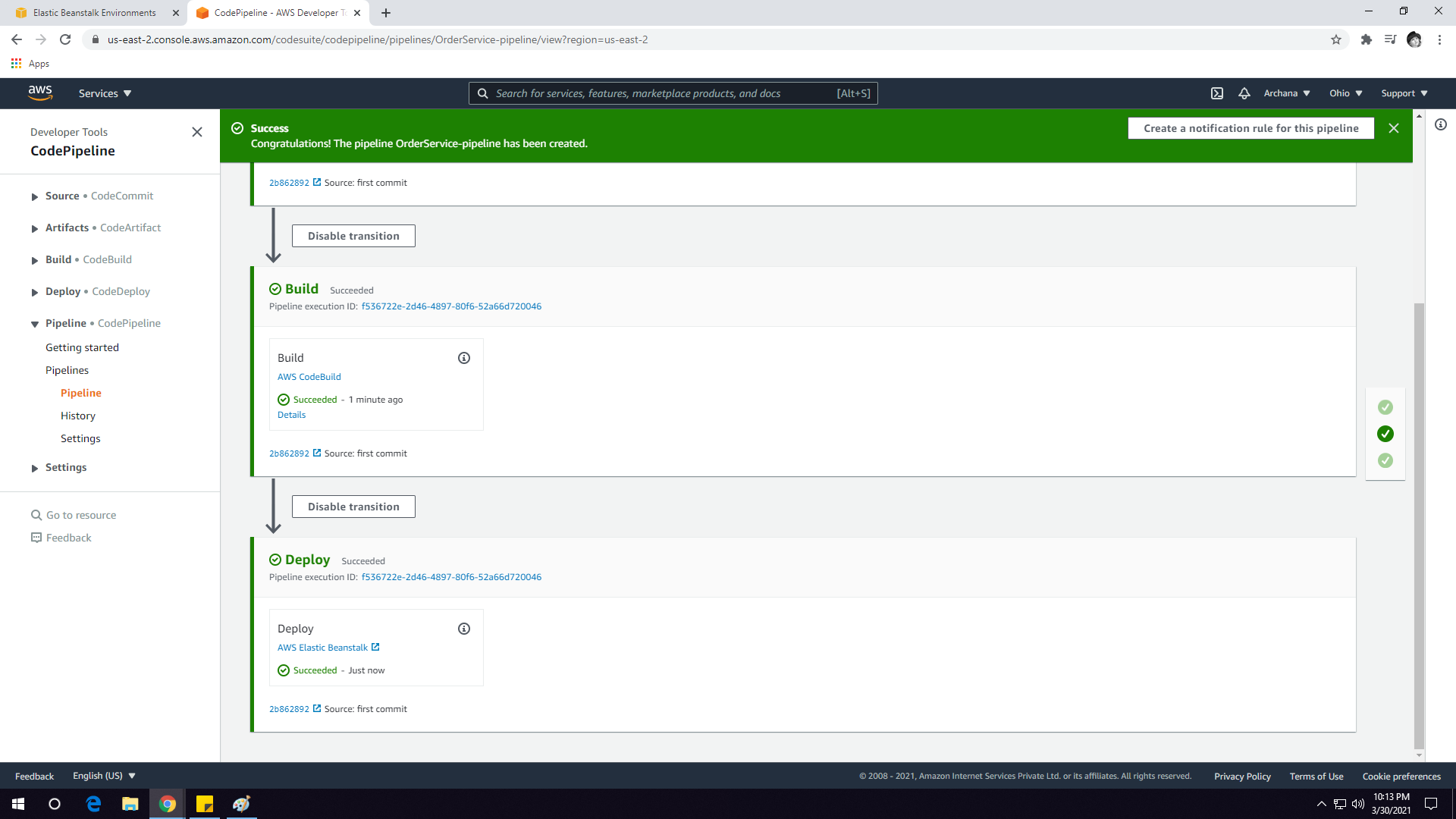
Deploying the pipeline with Elastic beanstalk environment created earlier



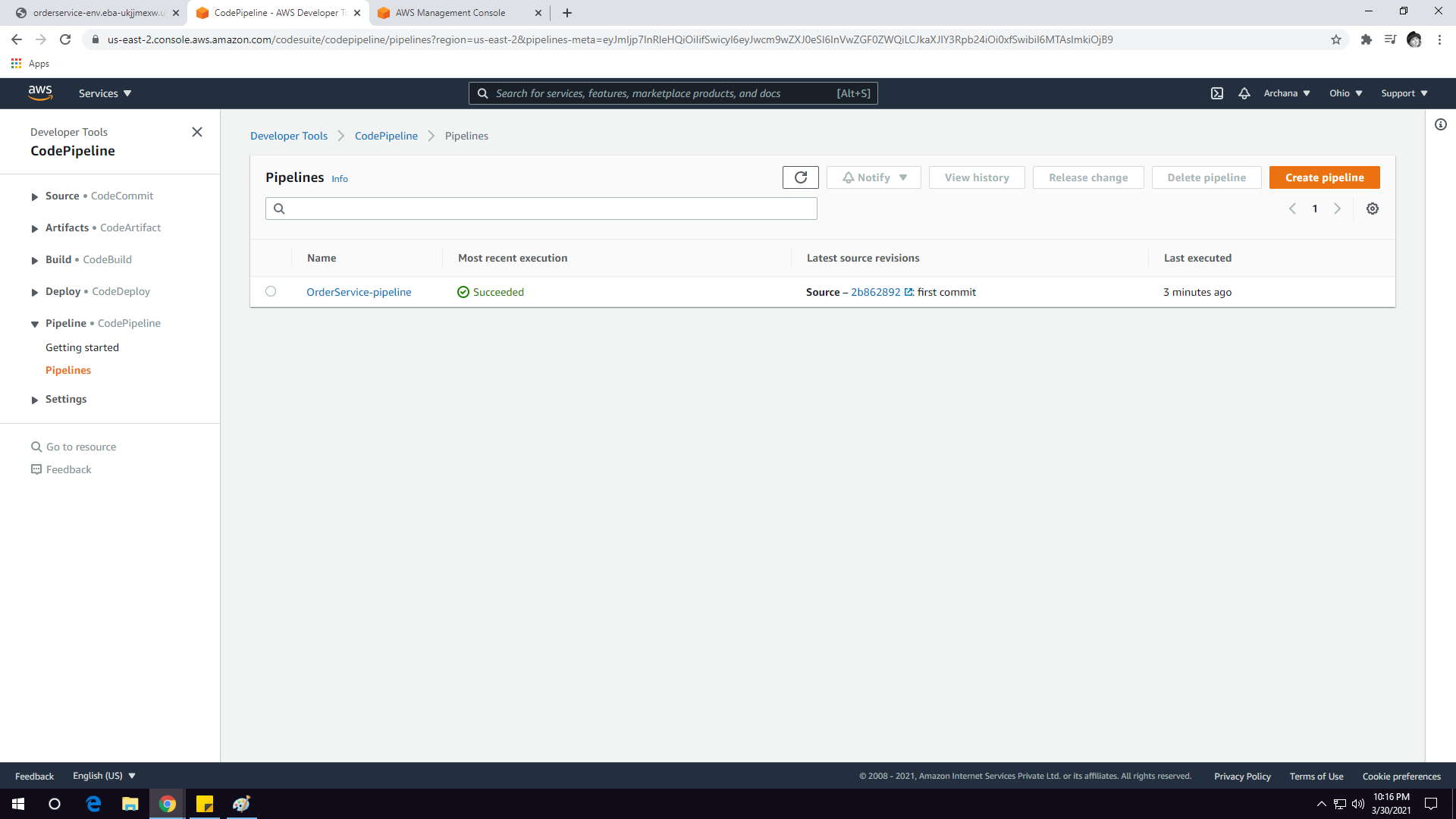
Pipeline has been created

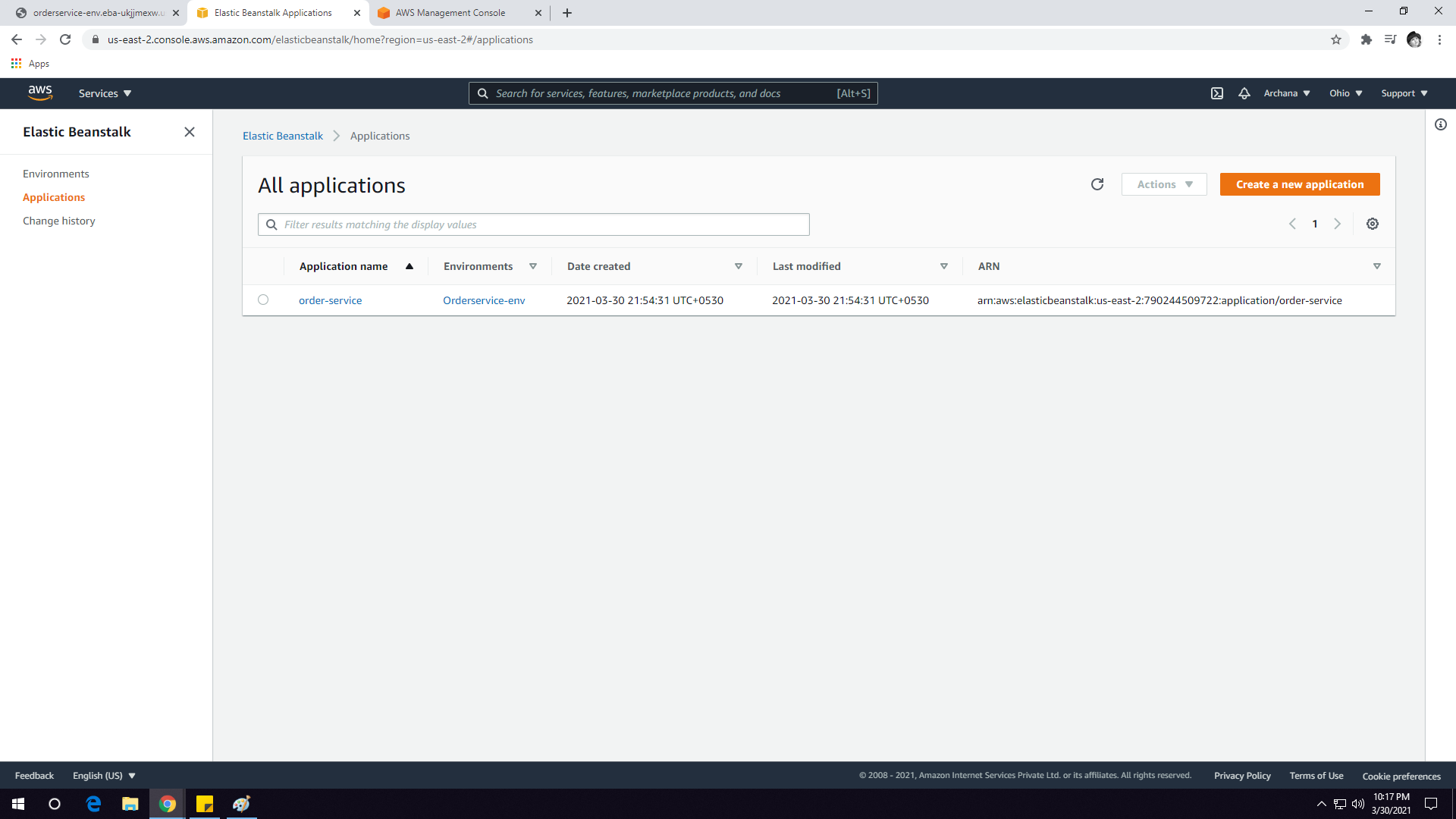


Deployed successfully



Viewing/Checking the status of pipeline



Beanstalk Application is up and running

Final Output

