🚀 Cloud DevOps Lab Adventure: Building an Automated Website Factory

Welcome, future cloud engineer! This guide will transform you into a DevOps wizard, teaching you how to build and automate websites in the cloud. Let's embark on this exciting journey!

🎯 Lab Mission Overview

The Big Picture: What We're Building

Imagine creating a magical factory that:

- **Takes your code** (like a recipe)
- **Automatically bakes it** into a website (like a chef)
- **Instantly serves it** to the world (like a super-fast delivery service)

Final Result: A website that says "Hello World" → "Hello World Change" automatically!

🏗️ Part 1: Setting Up Our Digital Workshop

1.1 🛠️ Presetting the Lab Environment

Clicked “Preset Lab Environment”

Got a pre-built digital workshop

Like getting a ready-to-use kitchen with all appliances installed

💡 **Key Concept**: **Cloud Resources**

- Think of these as **digital Lego blocks**:
  - **VPC (Virtual Private Cloud)**: Your private digital neighborhood
  - **Security Groups**: Digital security guards controlling who enters
  - **ECS (Elastic Cloud Server)**: Your virtual computer in the cloud

🎯 **Purpose**: We’re preparing our secure digital playground where all the magic happens!

1.2 🔐 Logging In to Huawei Cloud

Used lab credentials

Like using a practice driver’s car instead of your own

Not personal account

Keeps your real account safe and separate

🌟 **Pro Tip**: The **Exercise Interface** is your mission control center - everything happens here!

💻 Part 2: Renting a Cloud Computer (ECS)

2.1-2.2 🖥️ Creating Your Elastic Cloud Server

🚀 What is an ECS?

Imagine you need a computer that:

- ⏰ Runs 24/7 without stopping
- 🌐 Is always connected to the internet
- 💪 Can handle thousands of visitors
- 🔧 Can be resized anytime

That’s your **Elastic Cloud Server**!

📋 ECS Configuration Breakdown

```
# Our ECS Recipe:
Billing Mode: "Pay-per-use"          # Like paying for electricity - only for what you use!
Region: "AP-Singapore"               # Our computer lives in Singapore
CPU: "2 vCPUs"                       # 2 virtual brains
Memory: "4 GiB"                      # 4GB of thinking space
OS: "CentOS 7.6"                     # The computer's personality
Storage: "40 GB High I/O"            # Super-fast storage closet
Internet: "2 Mbit/s EIP"             # Our website's front door to the world
```

🔐 **Security Setup**:

- **VPC-hce**: Our private digital neighborhood
- **Security Group sg-hce**: Our digital bouncer deciding who gets in

🔑 **Login Credentials**:

```
Username: root       # The super-user (like being the boss of the computer)
Password: hC8iUA0UAs7V%hcj  # Our super-secret key to enter
```

🎯 **Achievement Unlocked**: 🏠 We now have a powerful cloud computer ready to host our website!

🏭 Part 3: Building Our Software Factory with CodeArts

3.1 📂 Creating a Project

What is a Project?
Think of it as your **project command center** - where you plan, track, and manage your website building.

Scrum Template: Like using a pre-made project planner that helps teams work together efficiently.
Scrum is a way for a team to work together on a big project by breaking it into small steps, talking about what’s working, and helping each other to finish.

3.2 📚 Creating a Code Repository

🌟 What’s a Code Repository?

Code Repository	A magical recipe book that remembers every change	Stores all your code and its history
Template	A pre-written recipe	Gives us a head start
“Java Web Demo”	A “Hello World” website recipe	Our starting point

🔮 Repository Settings:

- **Name**: WEB-Dev (our project’s name)
- **Visibility**: Private (only we can see it)
- **Permissions**: Who can read/write our recipe book

3.3 ✏️ Modifying the Code

The Change We Made:

```
// BEFORE: Website listens on the "side door"
server.port=8080

// AFTER: Website listens on the "main entrance"
server.port=80
```

🌐 **Why Port 80?**

- **Port 80**: The standard web door (http://)
- **Port 8080**: The developer’s testing door
- By using port 80, visitors don’t need to type “:8080” in the URL!

3.4 🔨 The Build Process - From Code to Program

🏗️ **Build Process Explained**:

```
Raw Code (Ingredients)
    → BUILD PROCESS (Robot Chef)
    → Runnable Program (Finished Cake)
```

What Happens During Build?:

1. **Compilation**: Checking if our code makes sense
2. **Packaging**: Putting all pieces together
3. **Testing**: Making sure everything works

4. **Creating Artifact**: The final, runnable program

3.5 🚀 Deployment: Going Live!

3.5.1 🎪 Configuring the Environment

Creating “web-env”: Our deployment playground name

Connecting to Our ECS:

```
Host Cluster: "group"           # A group of servers (we have one)
Connection Type: "Direct"       # Straight to our ECS
Host Details:
  - Name: "ecs-Linux"
  - IP: [Our ECS's public address]
  - Username: "root"
  - Password: "hC8iUA0uAs7V%hcj"
  - Port: 22 (SSH - secure connection door)
```

✅ **Connectivity Test**: Making sure CodeArts can talk to our ECS!

3.5.2 📋 Creating Deployment Instructions

Our Deployment Recipe:

```
# Deployment Parameters:
host_group: web-env              # Where to deploy
service_port: 80                 # Website door number
spring_path: /usr/local/expert/demoapp.jar  # Where to put our app
jdk_path: /usr/local/jdk         # Where Java lives
download_path: /usr/local/expert/ # Where to download files
```

🚀 **Deployment Steps**:

1. **Stop SpringBoot** (Skip for first time - nothing running yet!)
2. **Install JDK** (Java Development Kit - our website’s language interpreter)
3. **Download Build Artifact** (Get the finished website from build step)
4. **Start SpringBoot** (Launch our website!)

🎉 **The Magic Moment**:
After clicking “Save & Deploy”, we:

1. Visited our ECS IP address with “/test”
2. Saw **“hello world”** - OUR WEBSITE IS LIVE! 🌍

🔧 **Troubleshooting Tip**: If the website doesn’t show up, we might need to tell our security guard (Security Group) to allow visitors through door 80!

⚡ Part 4: DevOps Automation Magic!

4.1 🔄 Creating an Automated Pipeline

What is a Pipeline?
Imagine a **domino effect for software**:

```
Code Change → AUTOMATIC BUILD → AUTOMATIC TEST → AUTOMATIC DEPLOY → Website Updated!
```

Pipeline Configuration:

- **Trigger**: “Code commit” (when we save changes to our recipe book)
- **Result**: Entire process starts automatically!

4.2 ✍️ Making a Change

What We Changed:

```
// BEFORE:
return "hello world";

// AFTER:
return "hello world Change";
```

Commit Message: “Changed” - like leaving a note about what we modified in our recipe book.

4.3 🌟 Witnessing DevOps Magic!

The Automated Flow:

1. **We saved our code change** ✅

2. **Pipeline automatically detected the change** 🔍
3. **Build process started automatically** 🏗️
4. **Deployment process started automatically** 🚀
5. **Website updated automatically** ✨

The Proof: Refreshing our browser showed **"hello world Change"** without any manual steps!

🎓 Key Concepts Mastered

🏗️ DevOps Pipeline Explained

```
CODE → BUILD → TEST → DEPLOY → MONITOR
   ↻_________AUTOMATIC TRIGGER_________↻
```

🌐 Cloud Services We Used

ECS	Virtual Server	Renting a computer in the cloud
CodeArts	DevOps Platform	Automated software factory
VPC	Virtual Network	Private digital neighborhood
Security Group	Firewall Rules	Digital security guards
EIP	Public IP Address	Your website's street address

🔧 Technical Terms Demystified

- **Repository**: Magic recipe book that remembers everything
- **Build**: Turning code into a runnable program
- **Deploy**: Delivering and starting your program on a server
- **Pipeline**: Automated assembly line for software
- **DevOps**: Culture of automation and collaboration between developers and operations

---

🎉 Lab Achievement Unlocked!

You've successfully learned:

- ✅ How to create and configure cloud servers
- ✅ How to manage code using repositories
- ✅ How to build programs from source code
- ✅ How to deploy applications to production
- ✅ How to automate the entire software delivery process

🌟 **The Magic of DevOps**: You can now make changes to your website and have them automatically go live without manual intervention. This is how modern companies like Google, Netflix, and Amazon update their services!

🔮 Next Steps & Further Learning

What You Could Build Next:

- Personal portfolio website
- Blog application

- E-commerce store
- Mobile app backend

Advanced Concepts to Explore:

- **Kubernetes**: Managing multiple containers
- **Microservices**: Breaking apps into smaller pieces
- **CI/CD**: More advanced pipeline configurations
- **Cloud Monitoring**: Watching your application's health

💫 Final Wisdom

> "The power of DevOps is not just in the tools, but in the mindset of continuous improvement and automation. You've taken your first step into a larger world of cloud computing!"

Keep this guide handy - it's your roadmap to recalling everything you learned today. Happy coding! 🚀

📚 *Lab Notes Compiled with ❤️ for Future You*