

Cloud & Infrastructure as a Service (IaaS)

Key Takeaways

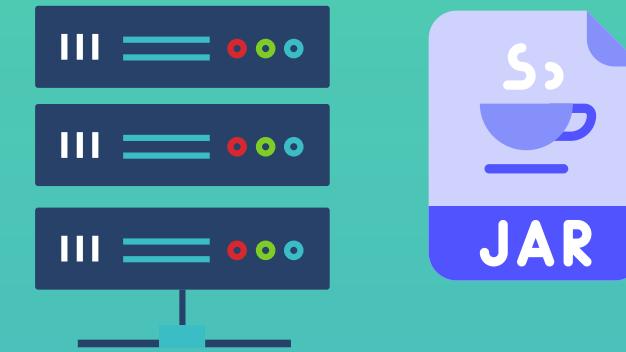
Throughout the DevOps Bootcamp, we will install and configure



Nexus
Artifact Repository



Jenkins
Build Automation



Deploy own
Applications on Servers

&
more

- We **don't install** it locally on our laptop!
- **Realistic way:** Install them on remote dedicated servers in cloud

What is Infrastructure as a Service (IaaS)?

- Offers **compute, storage and networking resources on demand**
- Instead of:

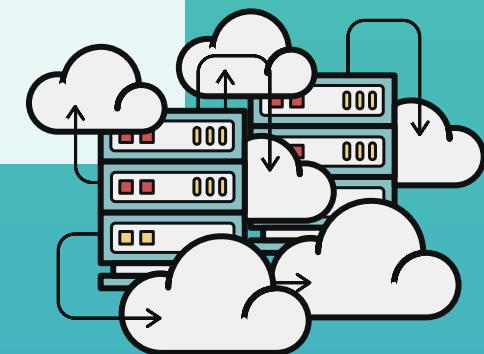
Company buys own server

- You manage own servers and infrastructure
- If something breaks, you need to fix it



Delegate Infrastructure Management

- Move your physical infrastructure to cloud
- You just rent the servers on demand



What is Cloud Computing?

- Cloud Computing is the **delivery of computing services** - including servers, storage, databases, networking, software - **over the internet** ("the cloud")
- **IaaS is 1 of 4 types of cloud services.** Others: Software as a Service (SaaS), Platform as a Service (PaaS) and Serverless

Infrastructure as a Service Providers

Most used IaaS Providers



- AWS most used one: **Powerful, but complex (learn in later module)**
- **DigitalOcean**: Easier to start



Setup Server on DigitalOcean (DO) - 1

to run packaged java application (jar file)

- Servers on DigitalOcean are called "Droplets"
- Droplets are Linux-based virtual machines (VMs)



Summary of Steps:

1. **Pre-Requisite:** Create DigitalOcean account (with new signup credits - free tier)
2. **Configure SSH keys**
3. **Create a Droplet** with Linux Ubuntu distribution
4. **Open SSH port 22** on server using firewall configuration
5. **SSH into the server** using its public IP address
6. **Install Java** to run Java applications on it

Setup Server on DigitalOcean (DO) - 2

to run packaged java application (jar file)

Configure SSH keys

- To be able to access any server on DO from local computer using SSH

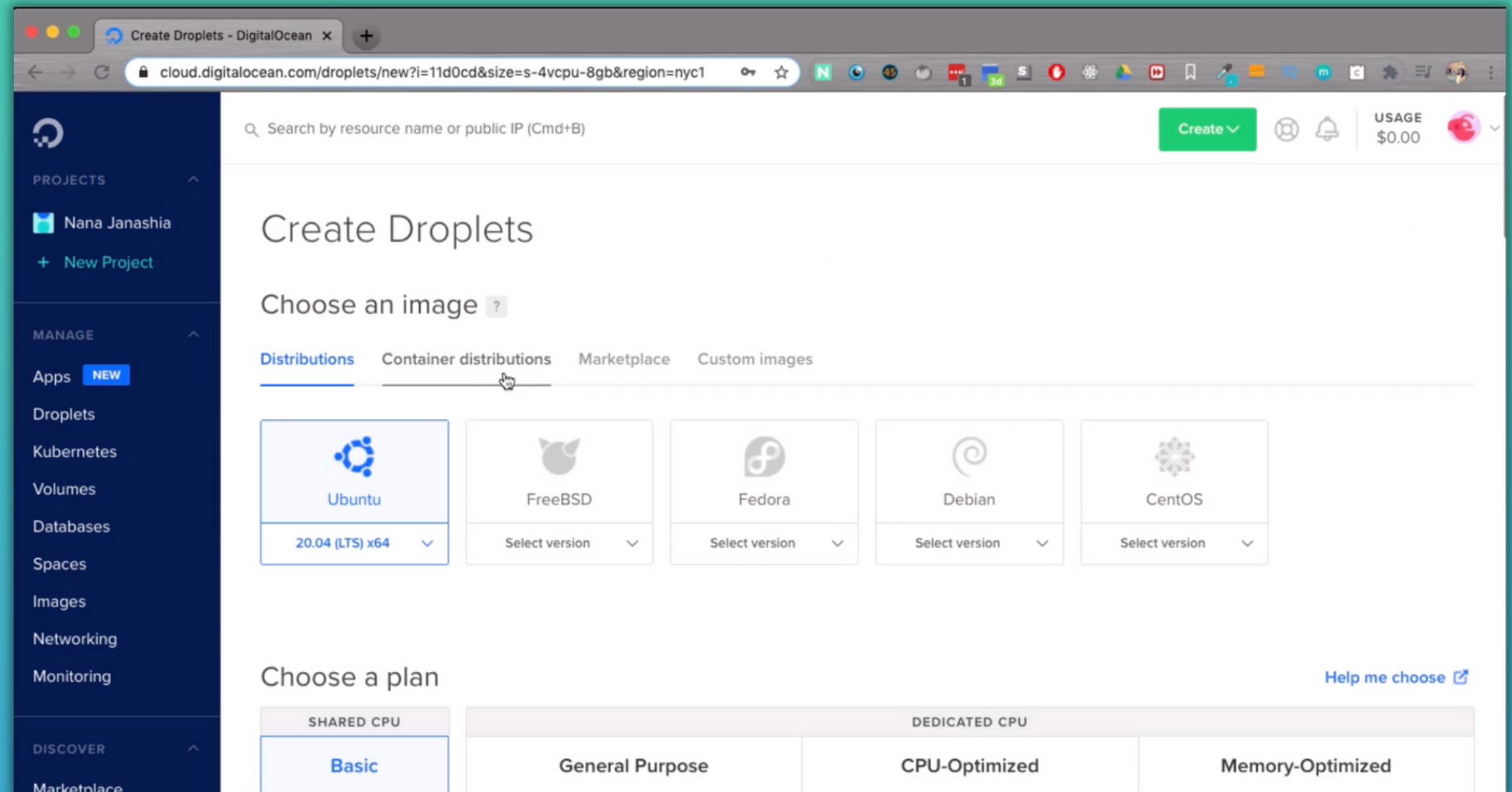
The screenshot shows the DigitalOcean 'Create Droplets' interface. On the left, there's a sidebar with 'PROJECTS' (Nana Janashia, New Project), 'MANAGE' (Apps NEW, Droplets, Kubernetes, Volumes, Databases, Spaces, Images, Networking, Monitoring), and 'DISCOVER' (Marketplace). The main area has tabs for 'Deploy multiple Droplets with the same configuration.' and 'Give your Droplets an identifying name you will remember them by. Your Droplet name can only contain alphanumeric characters, dashes, and periods.' A central modal window titled 'Add public SSH key' contains fields for 'SSH key content' (with a placeholder 'I') and 'Name' (with a placeholder '*'). Below these is a large 'Add SSH Key' button. To the right of the modal is a 'SSH Keys' panel with instructions for creating or adding SSH keys on Linux, MacOS & Windows, and for generating a new key pair using 'ssh-keygen'. It also includes a terminal command placeholder: 'Generating public/private rsa key pair. Enter file in which to save the key (/Users/USER/.ssh/id_rsa):'.

Setup Server on DigitalOcean (DO) - 3

to run packaged java application (jar file)

Create a Droplet

- With Linux Ubuntu distribution

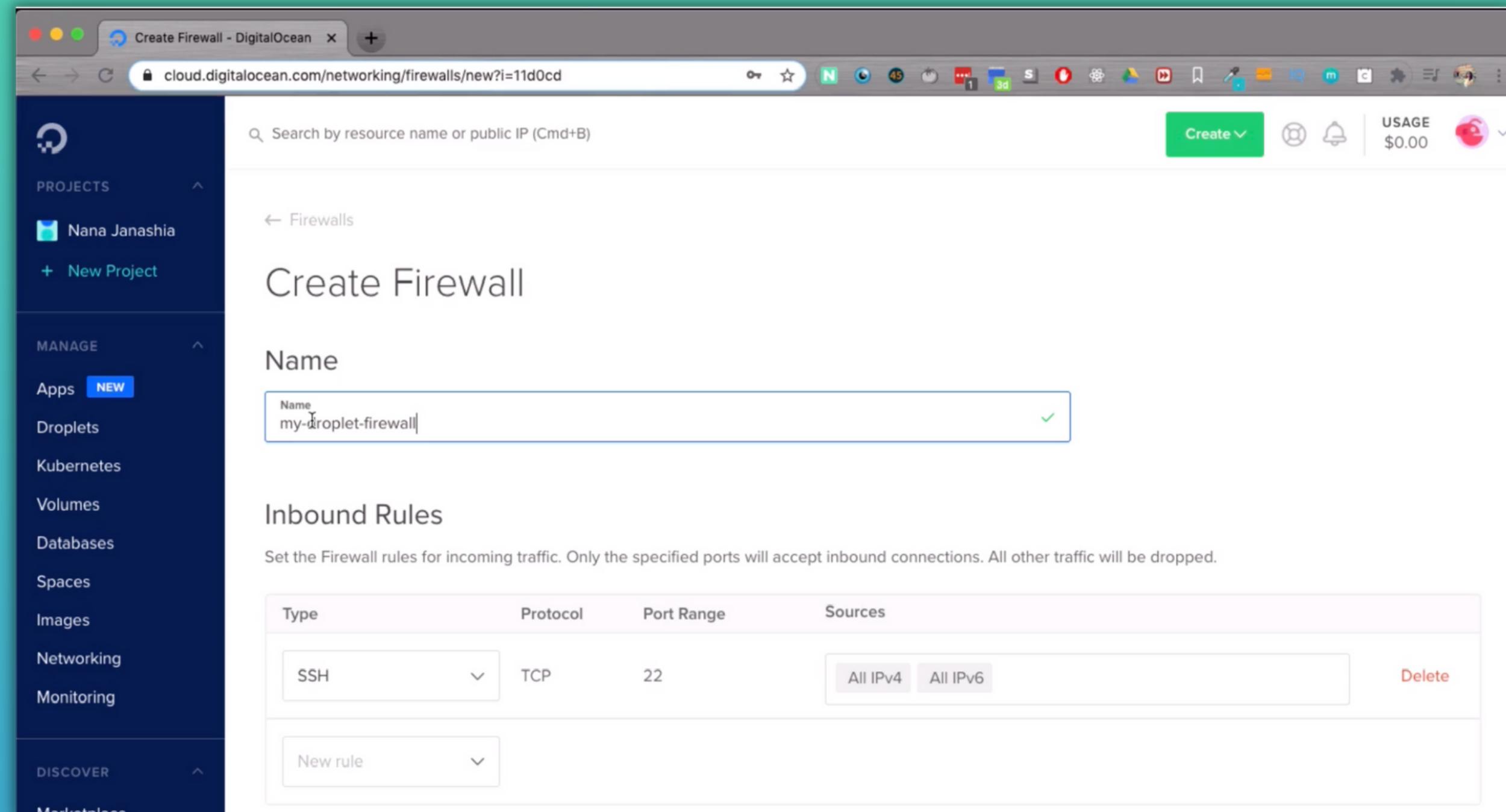


Setup Server on DigitalOcean (DO) - 4

to run packaged java application (jar file)

Open SSH port 22

- Open SSH port 22 on the server by creating a new **Firewall configuration**
- **Inbound Rules** = for incoming traffic
- **Outbound Rules** = for outgoing traffic

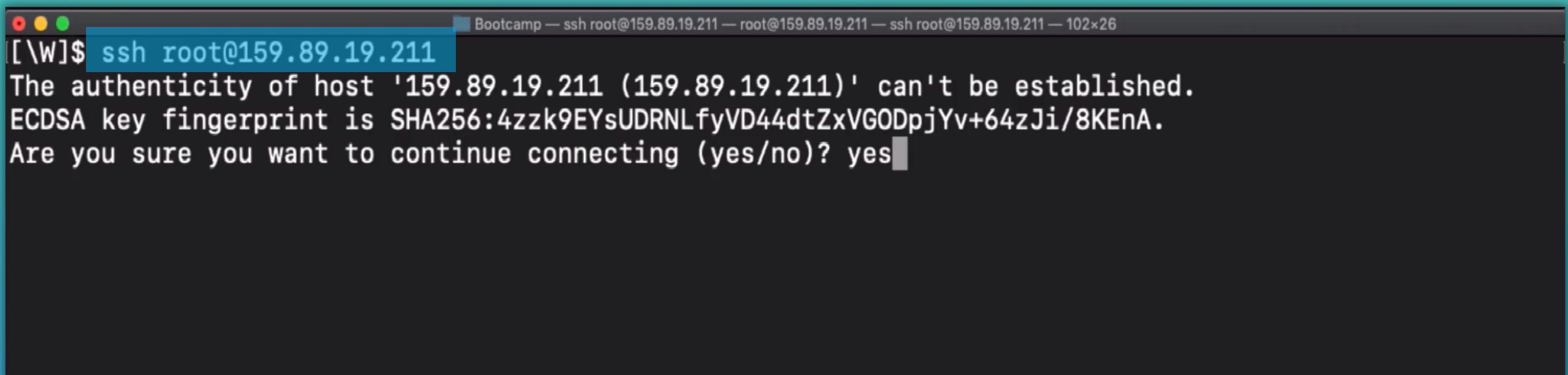


Setup Server on DigitalOcean (DO) - 5

to run packaged java application (jar file)

SSH into the server

- By using its public IP address



The screenshot shows a terminal window titled "Bootcamp — ssh root@159.89.19.211 — root@159.89.19.211 — ssh root@159.89.19.211 — 102x26". The command entered is "[\W]\$ ssh root@159.89.19.211". The terminal displays a warning message about host authenticity: "The authenticity of host '159.89.19.211 (159.89.19.211)' can't be established. ECDSA key fingerprint is SHA256:4zzk9EYsUDRNLfyVD44dtZxVGODpjYv+64zJi/8KEnA." It then asks the user if they are sure they want to continue connecting, with the options "yes/no". The user has typed "yes" and is awaiting a response.

Setup Server on DigitalOcean (DO) - 6

to run packaged java application (jar file)

Install Java

1. Check if java is installed and get installation commands



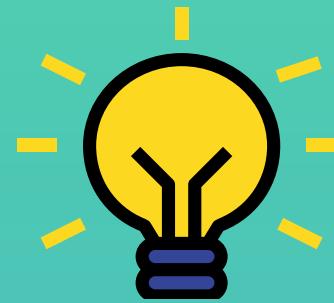
2. Install java to run Java applications on it

```
root@ubuntu-s-1vcpu-1gb-fra1-01:~# apt install openjdk-8-jre-headless
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java fontconfig-config fonts-dejavu-core java-common libavahi-client3
  libavahi-common-data libavahi-common3 libcups2 libfontconfig1 libjpeg-turbo8 libjpeg8 liblcms2-2
  libnspr4 libnss3 libpcsslite1 libxi6 libxrender1 libxtst6 x11-common
Suggested packages:
  default-jre cups-common liblcms2-utils pcscd libnss-mdns fonts-dejavu-extra fonts-ipafont-gothic
  fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zenhei fonts-indic
The following NEW packages will be installed:
  ca-certificates-java fontconfig-config fonts-dejavu-core java-common libavahi-client3
  libavahi-common-data libavahi-common3 libcups2 libfontconfig1 libjpeg-turbo8 libjpeg8 liblcms2-2
  libnspr4 libnss3 libpcsslite1 libxi6 libxrender1 libxtst6 openjdk-8-jre-headless x11-common
0 upgraded, 20 newly installed, 0 to remove and 16 not upgraded.
Need to get 30.7 MB of archives.
After this operation, 111 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

```
root@ubuntu-s-1vcpu-1gb-fra1-01:~# java
Command 'java' not found, but can be installed with:
apt install openjdk-11-jre-headless      # version 11.0.8+10-0ubuntu1~20.04, or
apt install default-jre                  # version 2:1.11-72
apt install openjdk-8-jre-headless      # version 8u265-b01-0ubuntu2~20.04
apt install openjdk-13-jre-headless     # version 13.0.3+3-1ubuntu2
apt install openjdk-14-jre-headless     # version 14.0.1+7-1ubuntu1
root@ubuntu-s-1vcpu-1gb-fra1-01:~# apt update
```

Run packaged Java Application - 1

on prepared DigitalOcean server



In **real world**, applications will run on a remote server!



Server prepared to run java application jar

Summary of Steps:

1. Build jar file
2. Copy to remote server
3. Run the application

Run packaged Java Application - 2

on prepared DigitalOcean server

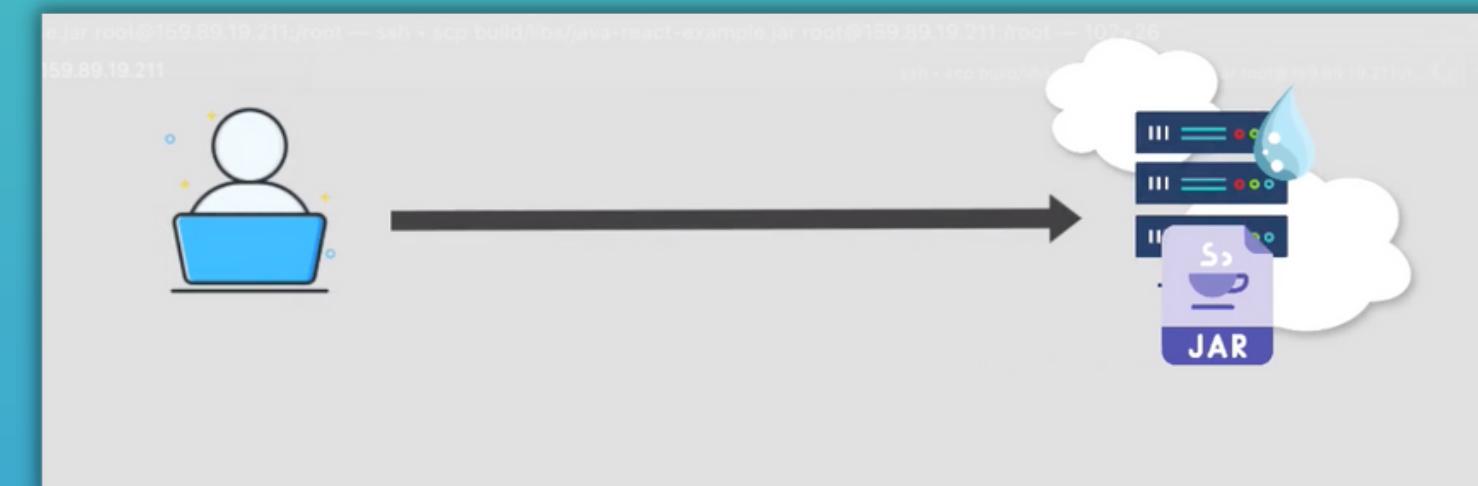
1. Build Jar File

```
[\W (master)]$ ./gradlew build
:compileJava
:compileGroovy NO-SOURCE
:processResources
:classes
:findMainClass
:jar
:bootRepackage
:assemble
:compileTestJava NO-SOURCE
:compileTestGroovy
:processTestResources NO-SOURCE
:testClasses
<===== 80% EXECUTING
> :test > 0 tests completed
```

2. Copy to Remote Server

```
:check
:build
BUILD SUCCESSFUL
Total time: 5.12 secs
[\W (master)]$ scp build/libs/java-react-example.jar root@159.89.19.211:/root
java-react-example.jar
```

jar file source locally remote server information
 |
 user public ip destination
 | | |
 45% 6400KB 2.4MB/s 00:03



Run packaged Java Application - 3

on prepared DigitalOcean server

3. Run the application

Security Best Practices

- **Create separate Linux user on remote server (as you learned in Linux module),**
 - Every cloud platform configuration for their remote servers is different
 - On droplet, per default you work with the "root" user
 - Create a new "admin" user
 - Using the "admin" user, create own users for each application you run (e.g. nexus, jenkins, my-app)
- Give it only the permission it needs to run that application
- **Don't work with the root user!**

