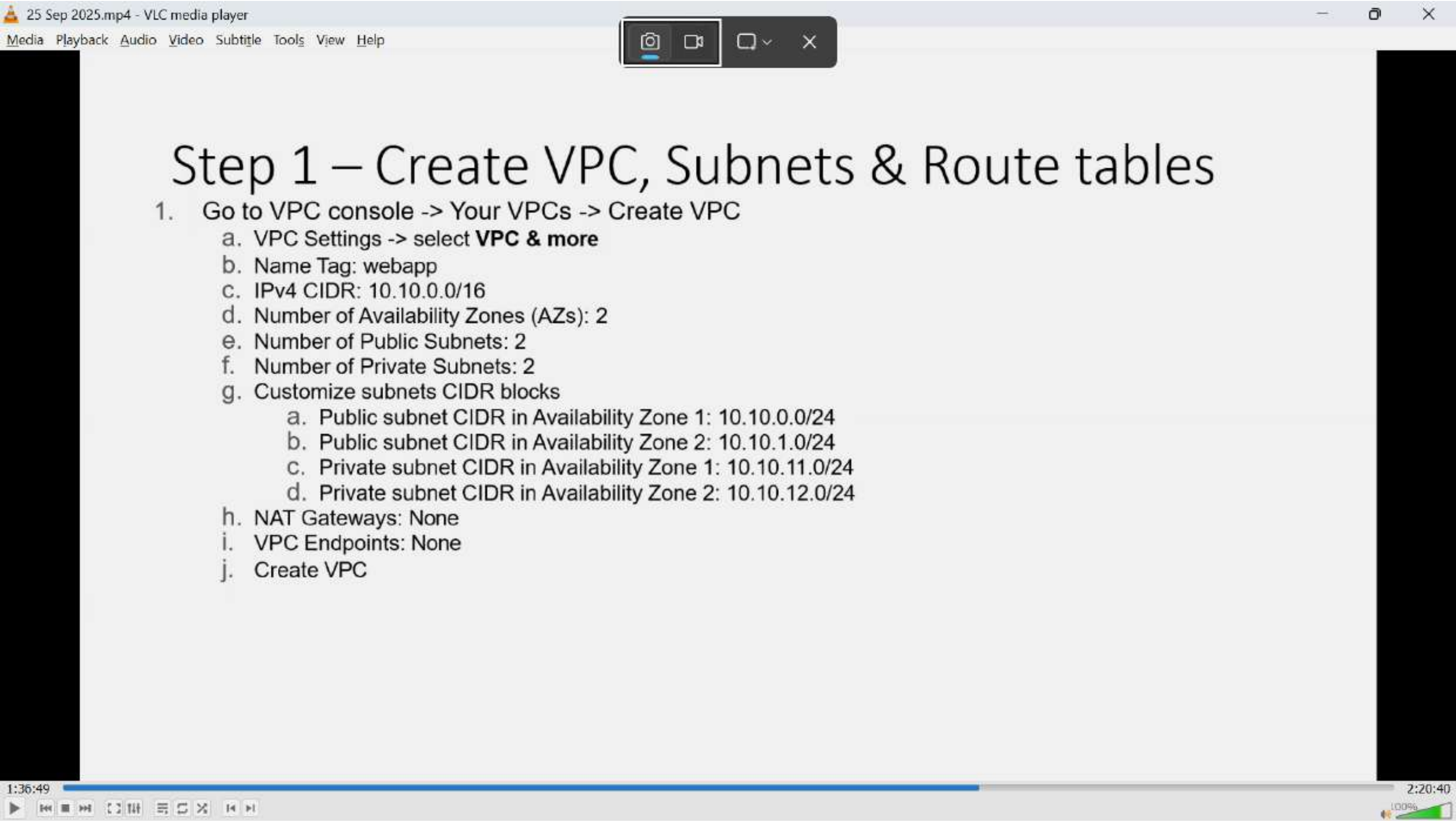
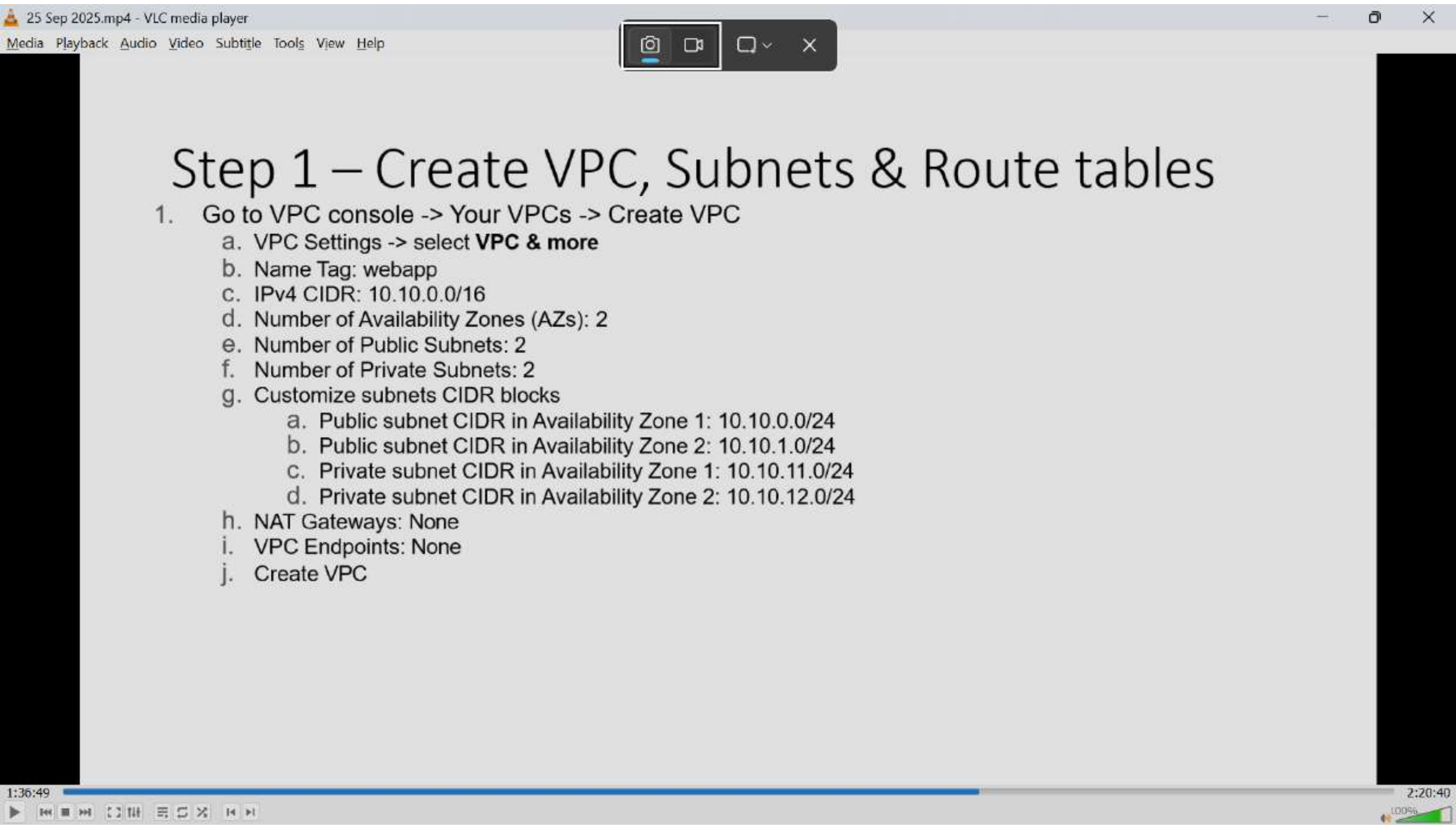


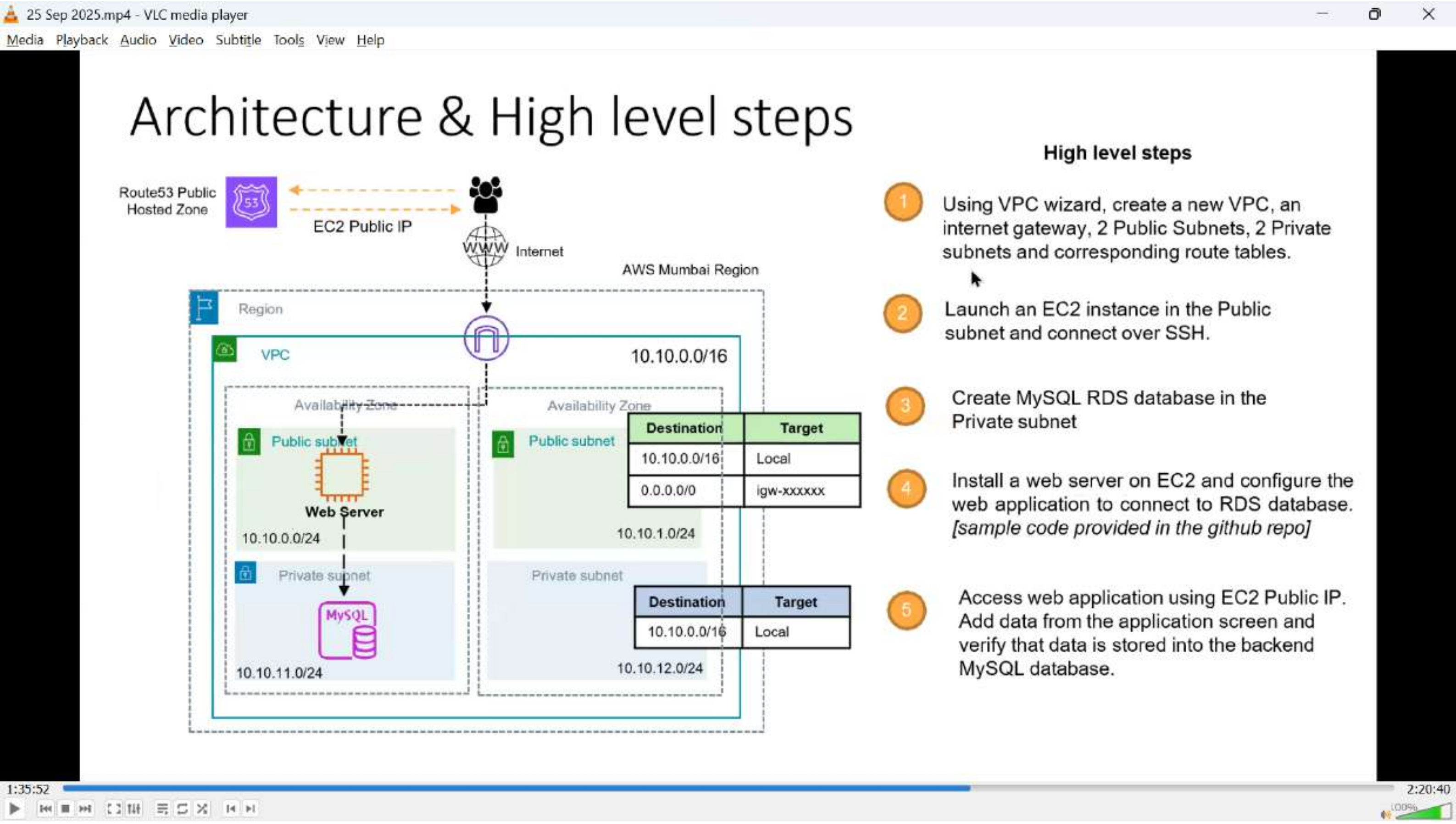
Step 1 – Create VPC, Subnets & Route tables

1. Go to VPC console -> Your VPCs -> Create VPC
   a. VPC Settings -> select **VPC & more**
   b. Name Tag: webapp
   c. IPv4 CIDR: 10.10.0.0/16
   d. Number of Availability Zones (AZs): 2
   e. Number of Public Subnets: 2
   f. Number of Private Subnets: 2
   g. Customize subnets CIDR blocks
      a. Public subnet CIDR in Availability Zone 1: 10.10.0.0/24
      b. Public subnet CIDR in Availability Zone 2: 10.10.1.0/24
      c. Private subnet CIDR in Availability Zone 1: 10.10.11.0/24
      d. Private subnet CIDR in Availability Zone 2: 10.10.12.0/24
   h. NAT Gateways: None
   i. VPC Endpoints: None
   j. Create VPC



Step 1 – Create VPC, Subnets & Route tables

1. Go to VPC console -> Your VPCs -> Create VPC
   a. VPC Settings -> select **VPC & more**
   b. Name Tag: webapp
   c. IPv4 CIDR: 10.10.0.0/16
   d. Number of Availability Zones (AZs): 2
   e. Number of Public Subnets: 2
   f. Number of Private Subnets: 2
   g. Customize subnets CIDR blocks
      a. Public subnet CIDR in Availability Zone 1: 10.10.0.0/24
      b. Public subnet CIDR in Availability Zone 2: 10.10.1.0/24
      c. Private subnet CIDR in Availability Zone 1: 10.10.11.0/24
      d. Private subnet CIDR in Availability Zone 2: 10.10.12.0/24
   h. NAT Gateways: None
   i. VPC Endpoints: None
   j. Create VPC


25 Sep 2025.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
1:35:52
2:20:40
100%

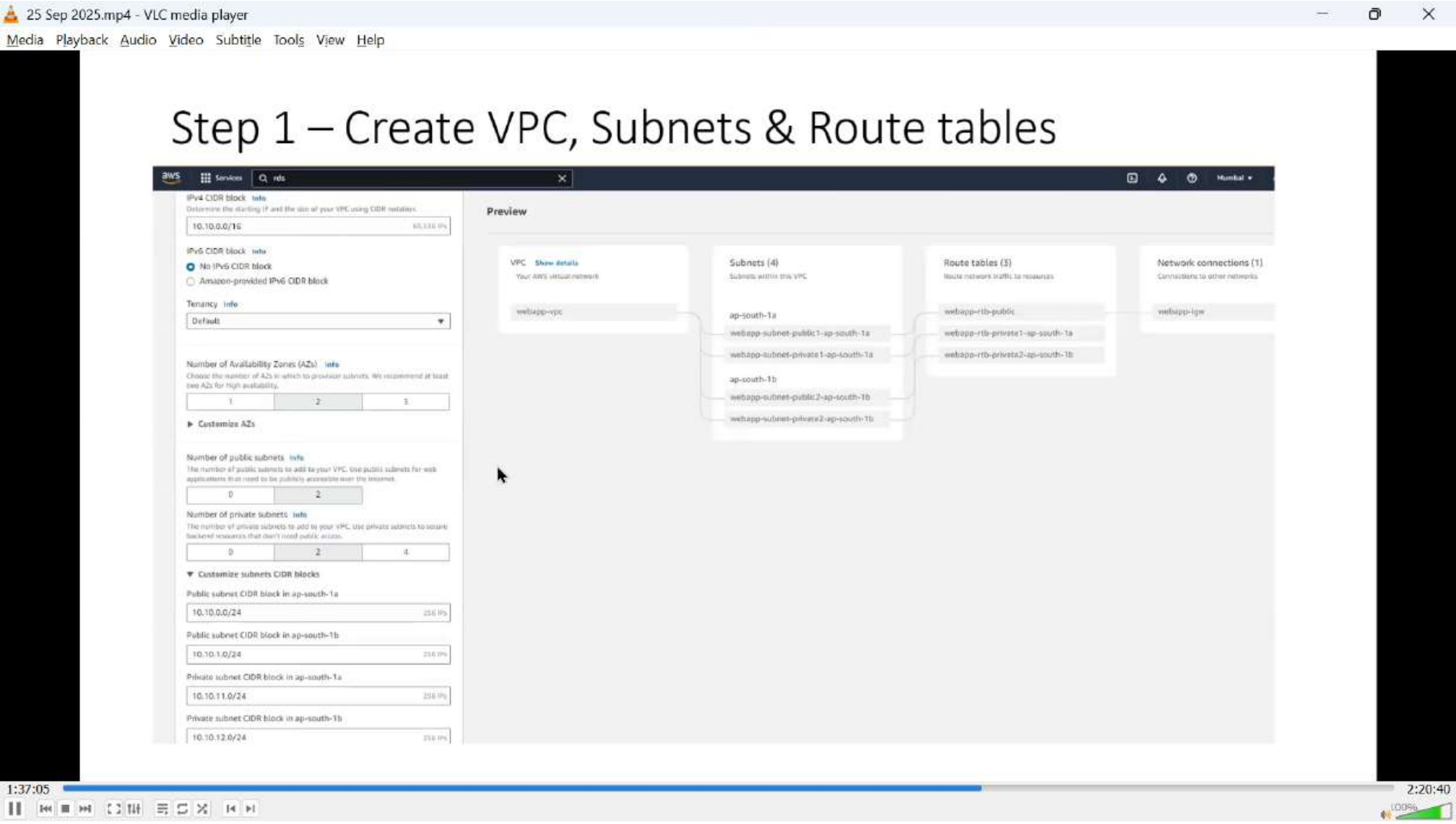
Architecture & High level steps

Destination	Target
10.10.0.0/16	Local
0.0.0.0/0	igw-xxxxxx

Destination	Target
10.10.0.0/16	Local

High level steps

1. Using VPC wizard, create a new VPC, an internet gateway, 2 Public Subnets, 2 Private subnets and corresponding route tables.
2. Launch an EC2 instance in the Public subnet and connect over SSH.
3. Create MySQL RDS database in the Private subnet
4. Install a web server on EC2 and configure the web application to connect to RDS database. *[sample code provided in the github repo]*
5. Access web application using EC2 Public IP. Add data from the application screen and verify that data is stored into the backend MySQL database.



Step 1 – Create VPC, Subnets & Route tables

IPv4 CIDR block Info
Determine the starting IP and the size of your VPC using CIDR notation.
10.10.0.0/16 65,536 IPs

IPv6 CIDR block Info
● No IPv6 CIDR block
○ Amazon-provided IPv6 CIDR block

Tenancy Info
Default ▾

Number of Availability Zones (AZs) Info
Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.
1 | [✓ 2] | 3

▶ Customize AZs

Number of public subnets Info
The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.
0 | [✓ 2]

Number of private subnets Info
The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.
0 | [✓ 2] | 4

▼ Customize subnets CIDR blocks

Public subnet CIDR block in ap-south-1a
10.10.0.0/24 256 IPs

Public subnet CIDR block in ap-south-1b
10.10.1.0/24 256 IPs

Private subnet CIDR block in ap-south-1a
10.10.11.0/24 256 IPs

Private subnet CIDR block in ap-south-1b
10.10.12.0/24 256 IPs

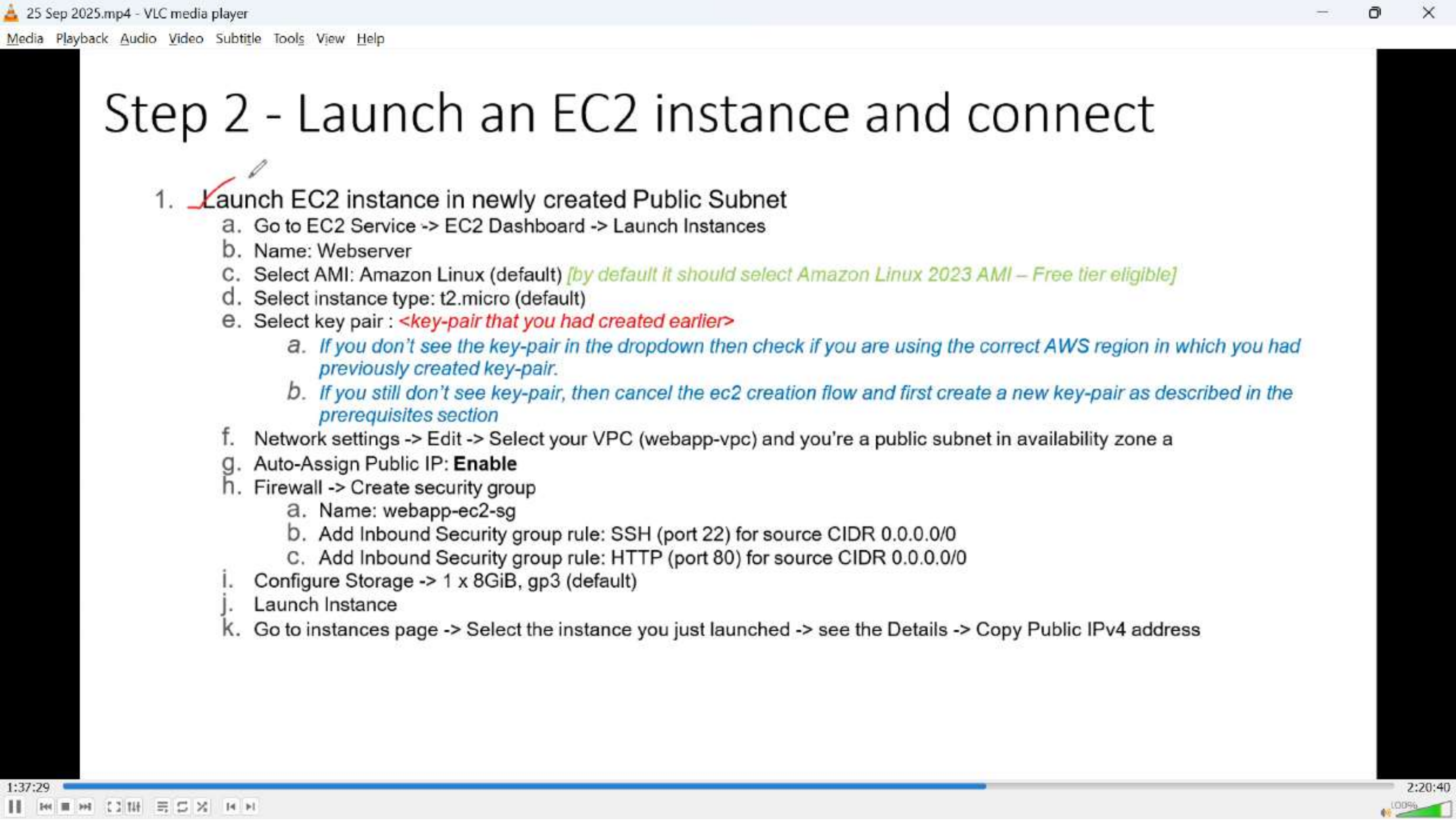
Preview

VPC Show details
Your AWS virtual network
webapp-vpc

Subnets (4)
Subnets within this VPC
ap-south-1a
webapp-subnet-public1-ap-south-1a
webapp-subnet-private1-ap-south-1a
ap-south-1b
webapp-subnet-public2-ap-south-1b
webapp-subnet-private2-ap-south-1b

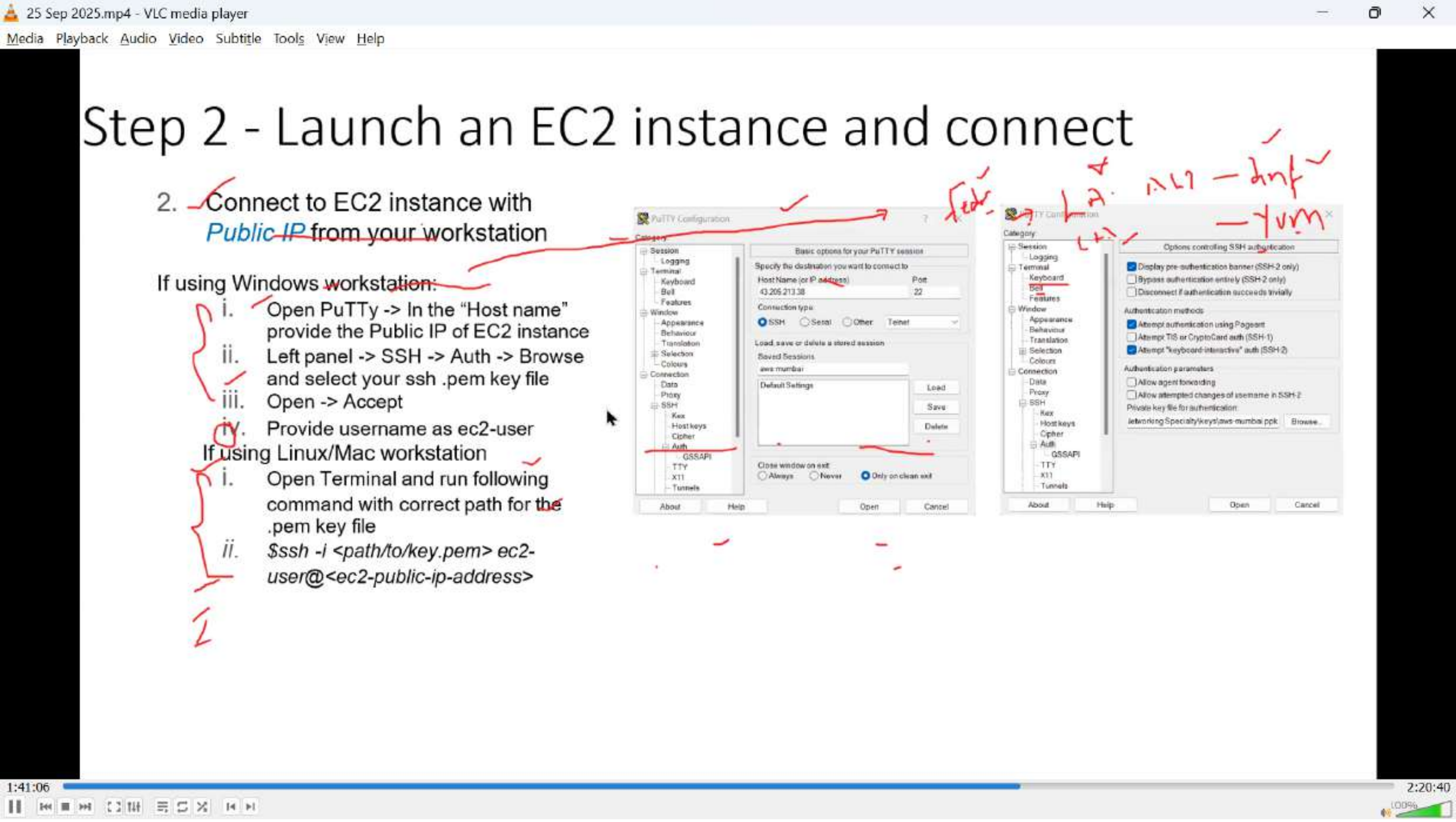
Route tables (3)
Route network traffic to resources
webapp-rtb-public
webapp-rtb-private1-ap-south-1a
webapp-rtb-private2-ap-south-1b

Network connections (1)
Connections to other networks
webapp-igw


25 Sep 2025.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
1:37:29
2:20:40
100%

Step 2 - Launch an EC2 instance and connect

1. Launch EC2 instance in newly created Public Subnet
   a. Go to EC2 Service -> EC2 Dashboard -> Launch Instances
   b. Name: Webserver
   c. Select AMI: Amazon Linux (default) *[by default it should select Amazon Linux 2023 AMI – Free tier eligible]*
   d. Select instance type: t2.micro (default)
   e. Select key pair : *<key-pair that you had created earlier>*
      a. *If you don't see the key-pair in the dropdown then check if you are using the correct AWS region in which you had previously created key-pair.*
      b. *If you still don't see key-pair, then cancel the ec2 creation flow and first create a new key-pair as described in the prerequisites section*
   f. Network settings -> Edit -> Select your VPC (webapp-vpc) and you're a public subnet in availability zone a
   g. Auto-Assign Public IP: **Enable**
   h. Firewall -> Create security group
      a. Name: webapp-ec2-sg
      b. Add Inbound Security group rule: SSH (port 22) for source CIDR 0.0.0.0/0
      c. Add Inbound Security group rule: HTTP (port 80) for source CIDR 0.0.0.0/0
   i. Configure Storage -> 1 x 8GiB, gp3 (default)
   j. Launch Instance
   k. Go to instances page -> Select the instance you just launched -> see the Details -> Copy Public IPv4 address


25 Sep 2025.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
1:41:06
2:20:40
100%

Step 2 - Launch an EC2 instance and connect

2. Connect to EC2 instance with *Public IP* from your workstation

If using Windows workstation:

- i. Open PuTTy -> In the "Host name" provide the Public IP of EC2 instance
- ii. Left panel -> SSH -> Auth -> Browse and select your ssh .pem key file
- iii. Open -> Accept
- iv. Provide username as ec2-user

If using Linux/Mac workstation

- i. Open Terminal and run following command with correct path for the .pem key file
- *ii. $ssh -i <path/to/key.pem> ec2-user@<ec2-public-ip-address>*

PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 43.205.213.38
Port: 22

Connection type: ● SSH ○ Serial ○ Other Telnet

Load, save or delete a stored session

Saved Sessions: aws mumbai

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

PuTTY Configuration

Options controlling SSH authentication

☑ Display pre-authentication banner (SSH-2 only)
□ Bypass authentication entirely (SSH-2 only)
□ Disconnect if authentication succeeds trivially

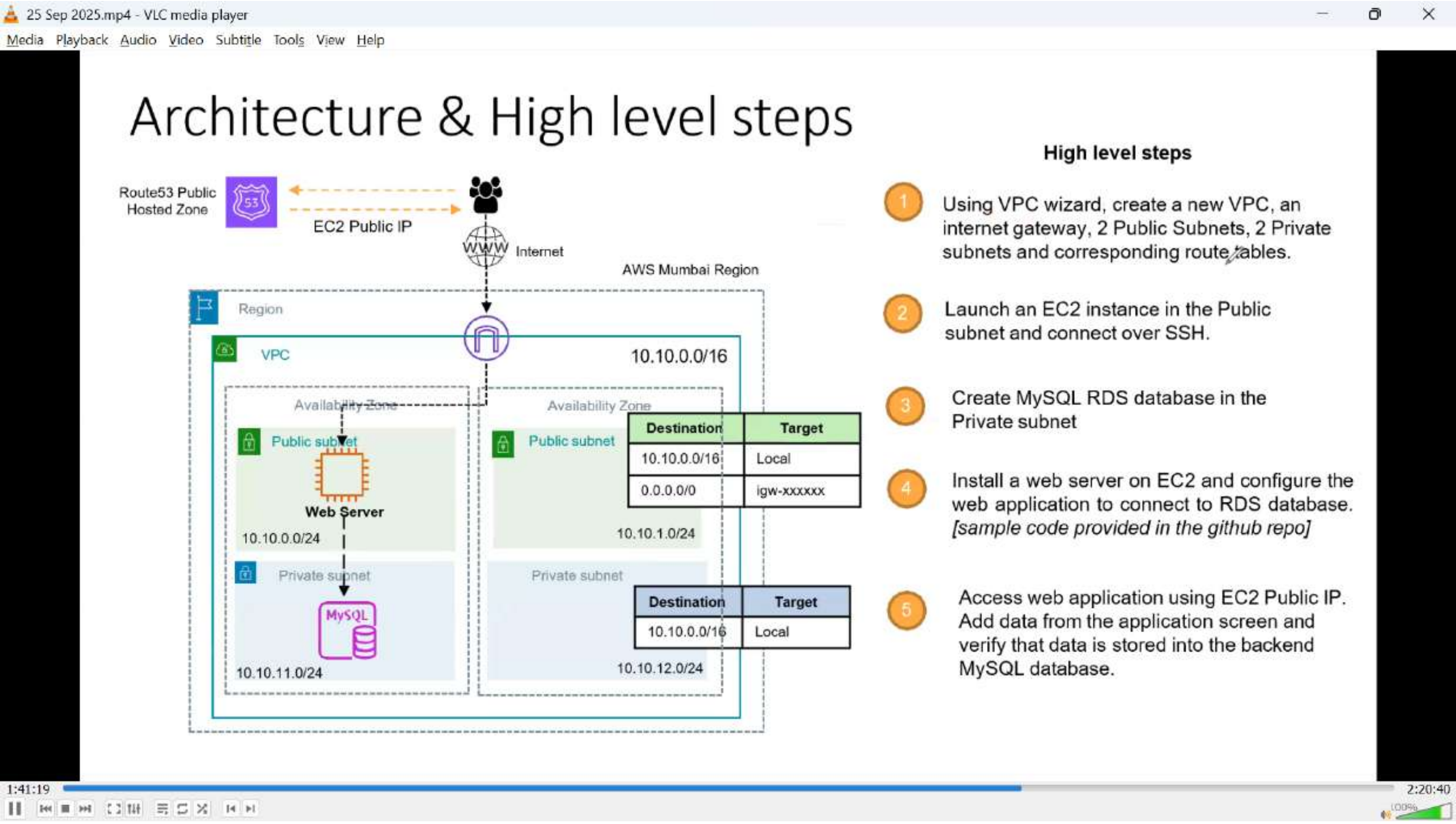
Authentication methods

☑ Attempt authentication using Pageant
□ Attempt TIS or CryptoCard auth (SSH-1)
☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

□ Allow agent forwarding
□ Allow attempted changes of username in SSH-2

Private key file for authentication: letworking Specialty\keys\aws-mumbai.ppk



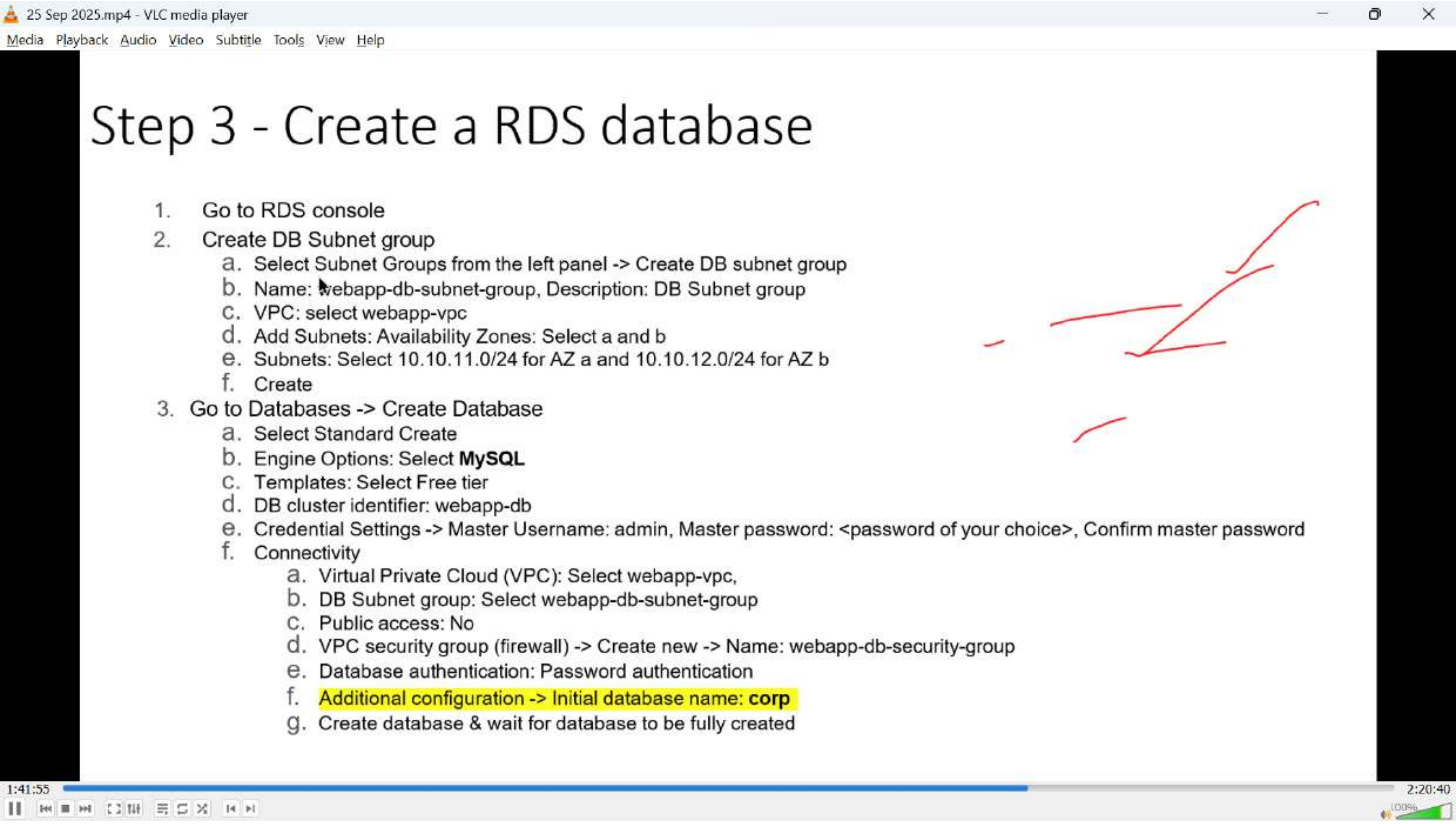
Architecture & High level steps

Destination	Target
10.10.0.0/16	Local
0.0.0.0/0	igw-xxxxxx

Destination	Target
10.10.0.0/16	Local

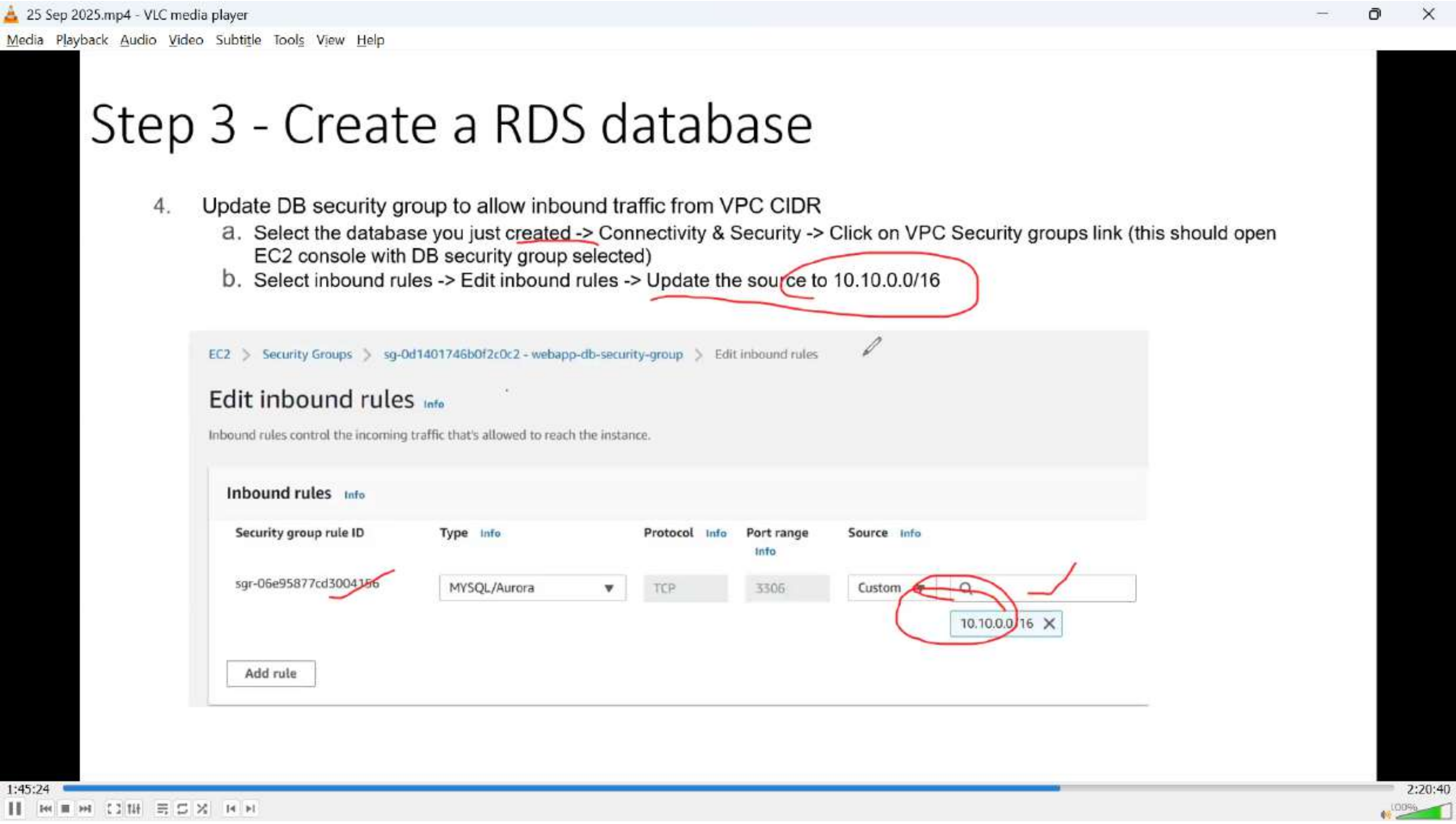
High level steps

1. Using VPC wizard, create a new VPC, an internet gateway, 2 Public Subnets, 2 Private subnets and corresponding route tables.
2. Launch an EC2 instance in the Public subnet and connect over SSH.
3. Create MySQL RDS database in the Private subnet
4. Install a web server on EC2 and configure the web application to connect to RDS database. *[sample code provided in the github repo]*
5. Access web application using EC2 Public IP. Add data from the application screen and verify that data is stored into the backend MySQL database.



Step 3 - Create a RDS database

1. Go to RDS console
2. Create DB Subnet group
   - a. Select Subnet Groups from the left panel -> Create DB subnet group
   - b. Name: webapp-db-subnet-group, Description: DB Subnet group
   - c. VPC: select webapp-vpc
   - d. Add Subnets: Availability Zones: Select a and b
   - e. Subnets: Select 10.10.11.0/24 for AZ a and 10.10.12.0/24 for AZ b
   - f. Create
3. Go to Databases -> Create Database
   - a. Select Standard Create
   - b. Engine Options: Select **MySQL**
   - c. Templates: Select Free tier
   - d. DB cluster identifier: webapp-db
   - e. Credential Settings -> Master Username: admin, Master password: <password of your choice>, Confirm master password
   - f. Connectivity
     - a. Virtual Private Cloud (VPC): Select webapp-vpc,
     - b. DB Subnet group: Select webapp-db-subnet-group
     - c. Public access: No
     - d. VPC security group (firewall) -> Create new -> Name: webapp-db-security-group
     - e. Database authentication: Password authentication
     - f. Additional configuration -> Initial database name: **corp**
     - g. Create database & wait for database to be fully created


25 Sep 2025.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help

Step 3 - Create a RDS database

4. Update DB security group to allow inbound traffic from VPC CIDR
   a. Select the database you just created -> Connectivity & Security -> Click on VPC Security groups link (this should open EC2 console with DB security group selected)
   b. Select inbound rules -> Edit inbound rules -> Update the source to 10.10.0.0/16

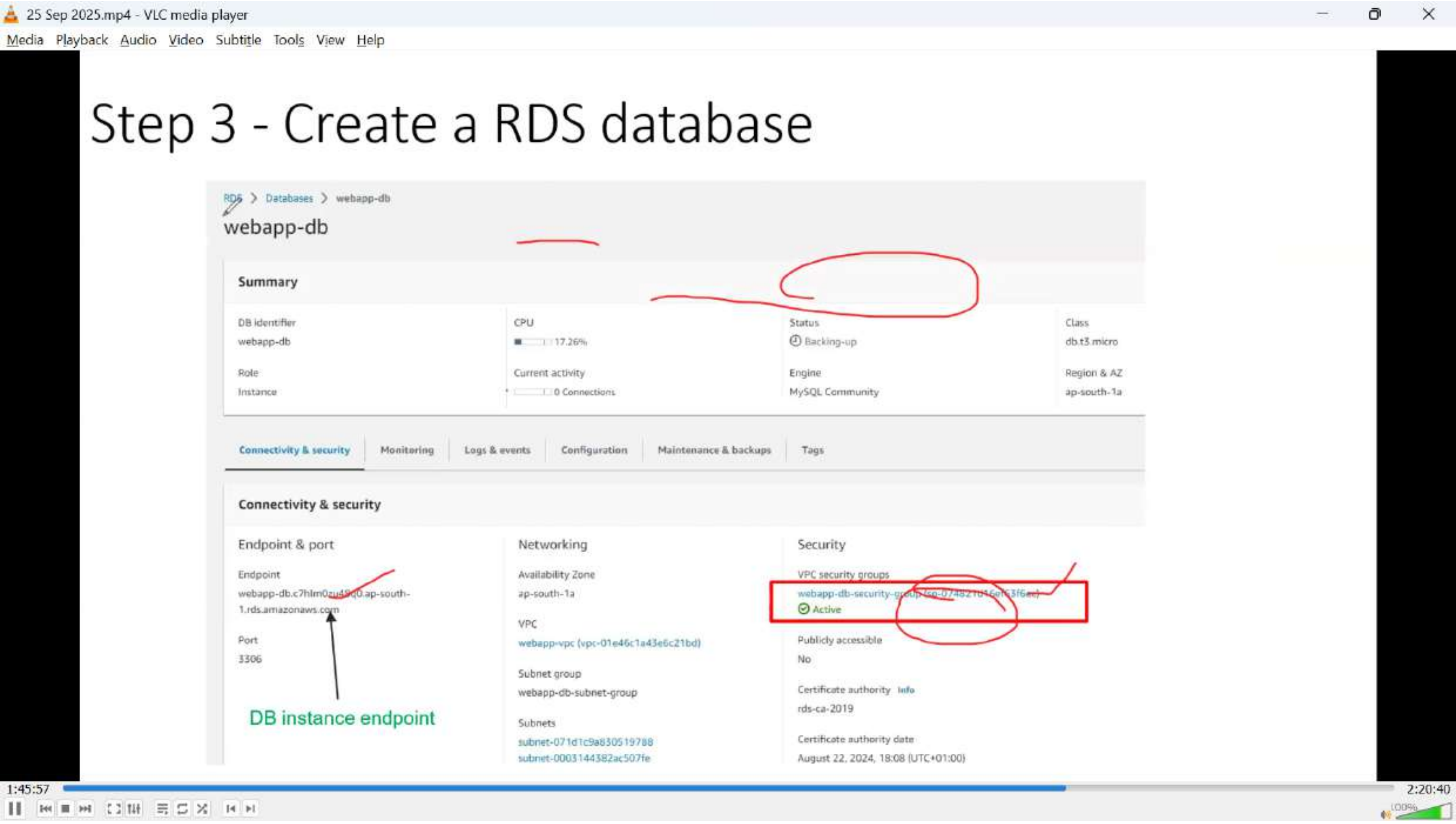
EC2 > Security Groups > sg-0d1401746b0f2c0c2 - webapp-db-security-group > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info
sgr-06e95877cd3004156	MYSQL/Aurora	TCP	3306	Custom 10.10.0.0/16



Step 3 - Create a RDS database

RDS > Databases > webapp-db

webapp-db

Summary

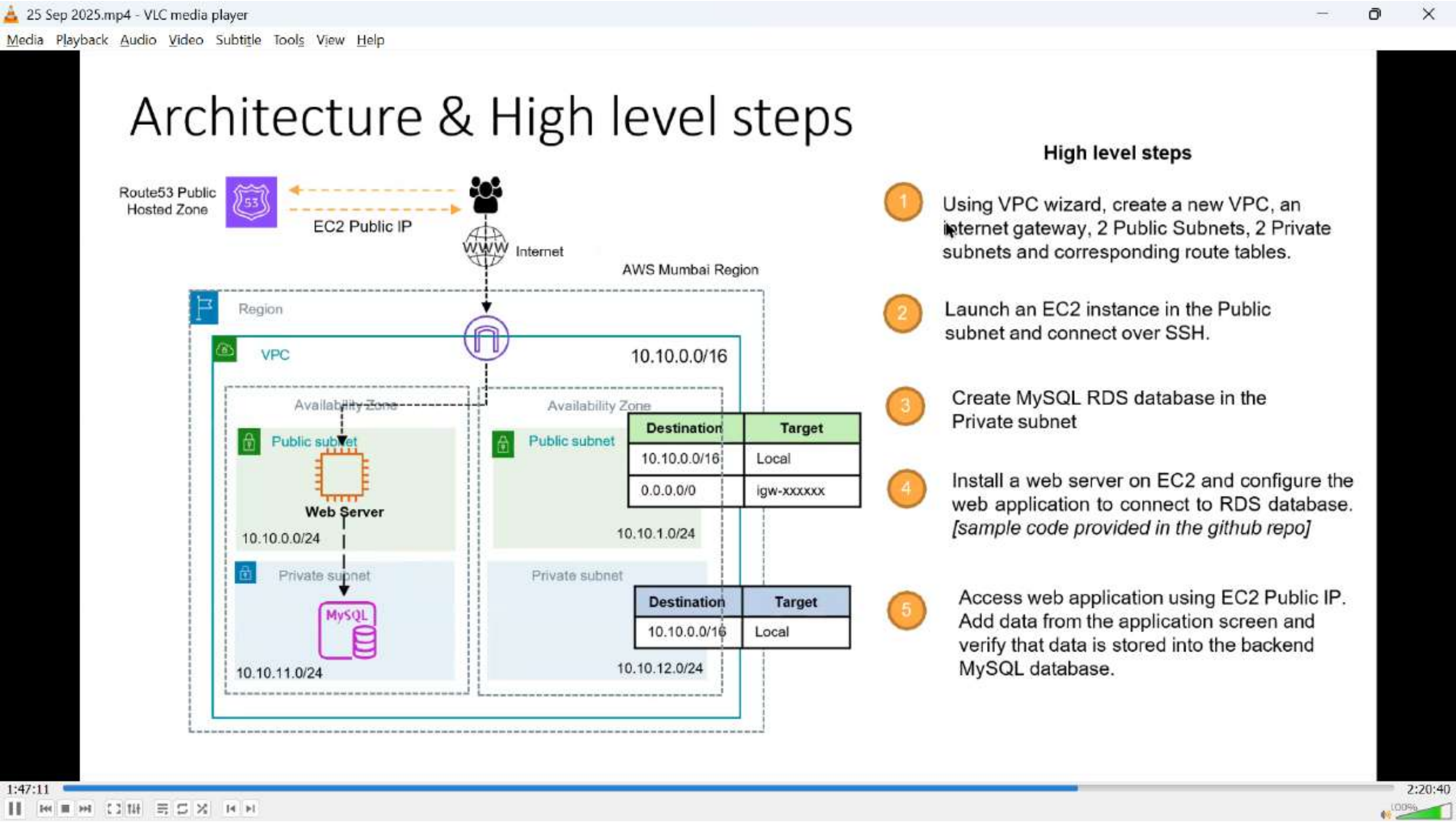
DB identifier	CPU	Status	Class
webapp-db	17.26%	Backing-up	db.t3.micro
Role	**Current activity**	**Engine**	**Region & AZ**
Instance	0 Connections	MySQL Community	ap-south-1a

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance & backups | Tags

Connectivity & security

Endpoint & port	Networking	Security
Endpoint webapp-db.c7hlm0zu48q0.ap-south-1.rds.amazonaws.com	Availability Zone ap-south-1a	VPC security groups webapp-db-security-group (sg-074821016ef53f6ae) Active
Port 3306	VPC webapp-vpc (vpc-01e46c1a43e6c21bd)	Publicly accessible No
	Subnet group webapp-db-subnet-group	Certificate authority Info rds-ca-2019
	Subnets subnet-071d1c9a830519788 subnet-0003144382ac507fe	Certificate authority date August 22, 2024, 18:08 (UTC+01:00)

DB instance endpoint


25 Sep 2025.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
1:47:11
2:20:40
100%

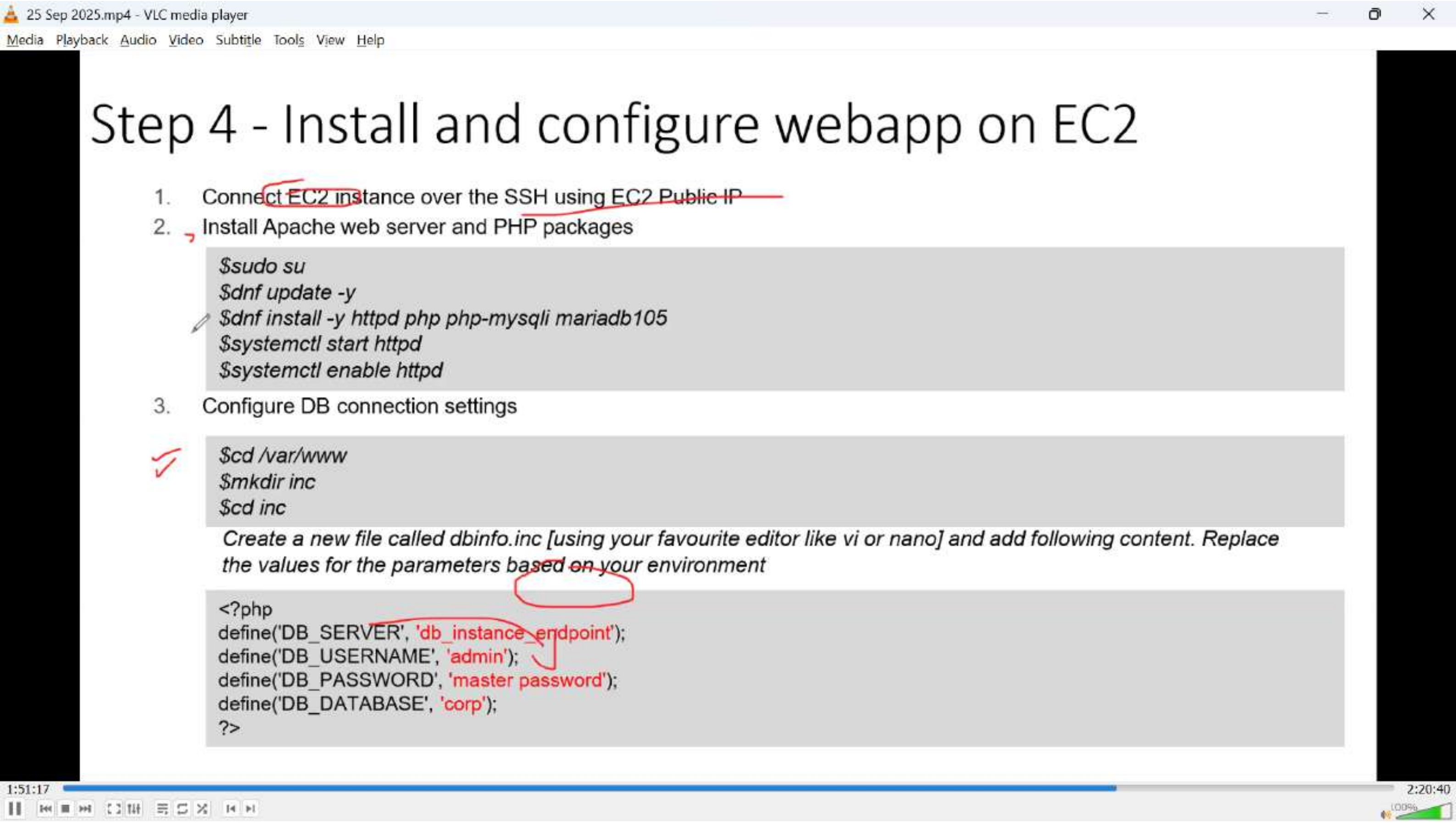
Architecture & High level steps

Destination	Target
10.10.0.0/16	Local
0.0.0.0/0	igw-xxxxxx

Destination	Target
10.10.0.0/16	Local

High level steps

1. Using VPC wizard, create a new VPC, an internet gateway, 2 Public Subnets, 2 Private subnets and corresponding route tables.
2. Launch an EC2 instance in the Public subnet and connect over SSH.
3. Create MySQL RDS database in the Private subnet
4. Install a web server on EC2 and configure the web application to connect to RDS database. *[sample code provided in the github repo]*
5. Access web application using EC2 Public IP. Add data from the application screen and verify that data is stored into the backend MySQL database.


25 Sep 2025.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
1:51:17
2:20:40
100%

Step 4 - Install and configure webapp on EC2

1. Connect EC2 instance over the SSH using EC2 Public IP
2. Install Apache web server and PHP packages

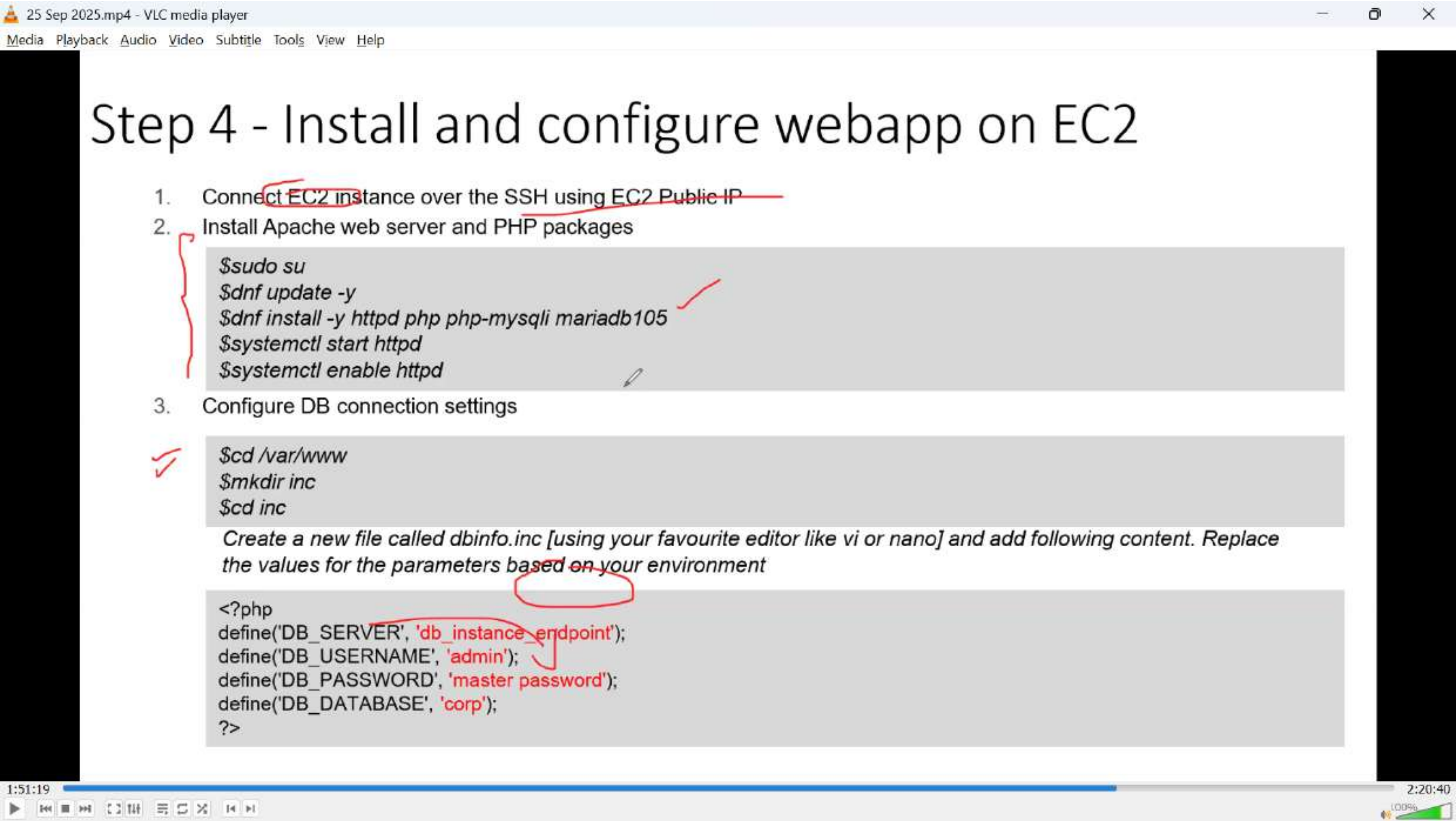
```
$sudo su
$dnf update -y
$dnf install -y httpd php php-mysqli mariadb105
$systemctl start httpd
$systemctl enable httpd
```

3. Configure DB connection settings

```
$cd /var/www
$mkdir inc
$cd inc
```

Create a new file called dbinfo.inc [using your favourite editor like vi or nano] and add following content. Replace the values for the parameters based on your environment

```
<?php
define('DB_SERVER', 'db_instance_endpoint');
define('DB_USERNAME', 'admin');
define('DB_PASSWORD', 'master password');
define('DB_DATABASE', 'corp');
?>
```


25 Sep 2025.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
1:51:19
2:20:40
100%

Step 4 - Install and configure webapp on EC2

1. Connect EC2 instance over the SSH using EC2 Public IP
2. Install Apache web server and PHP packages

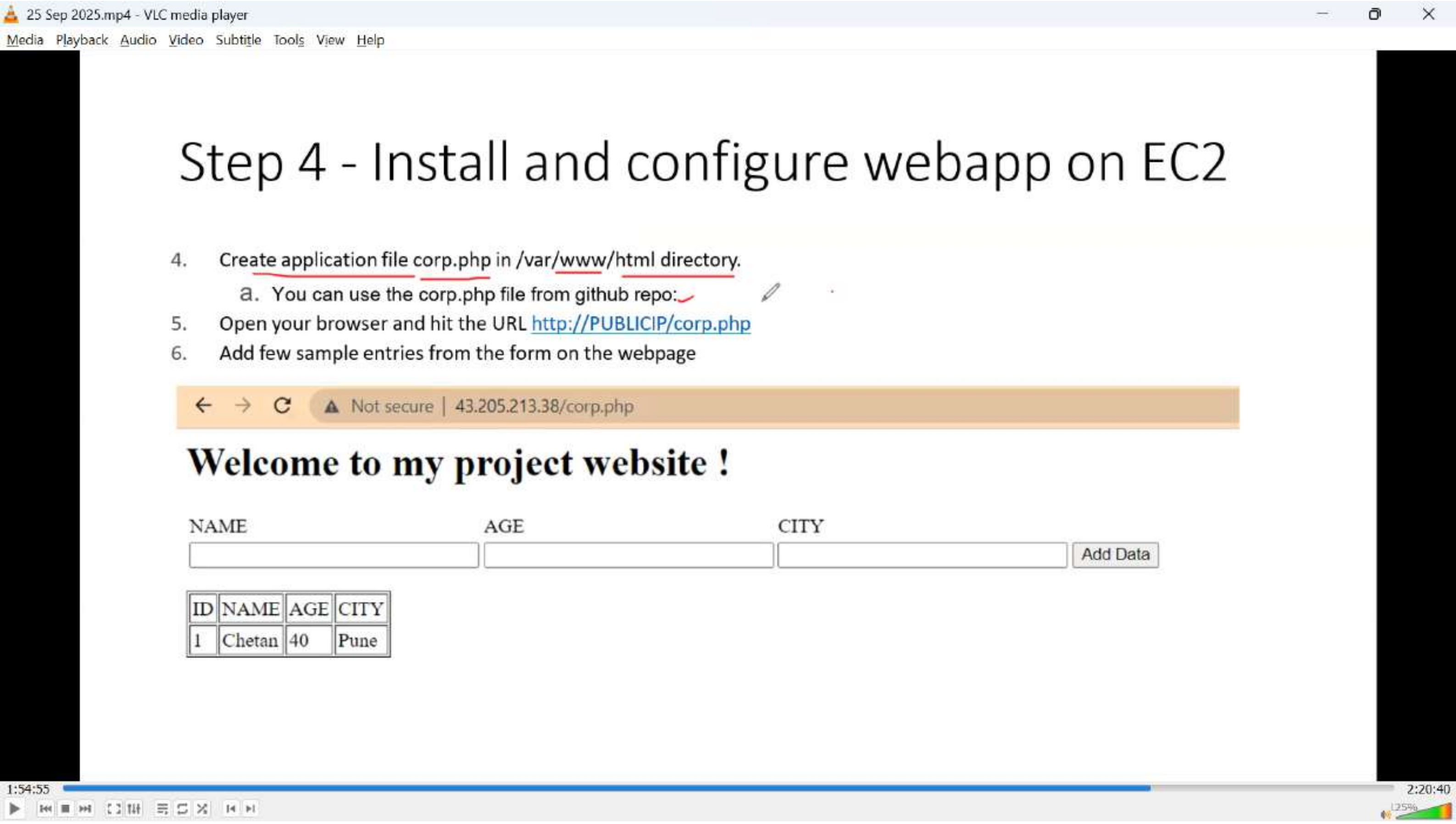
```
$sudo su
$dnf update -y
$dnf install -y httpd php php-mysqli mariadb105
$systemctl start httpd
$systemctl enable httpd
```

3. Configure DB connection settings

```
$cd /var/www
$mkdir inc
$cd inc
```

Create a new file called dbinfo.inc [using your favourite editor like vi or nano] and add following content. Replace the values for the parameters based on your environment

```
<?php
define('DB_SERVER', 'db_instance_endpoint');
define('DB_USERNAME', 'admin');
define('DB_PASSWORD', 'master password');
define('DB_DATABASE', 'corp');
?>
```


25 Sep 2025.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
1:54:55
2:20:40
125%

Step 4 - Install and configure webapp on EC2

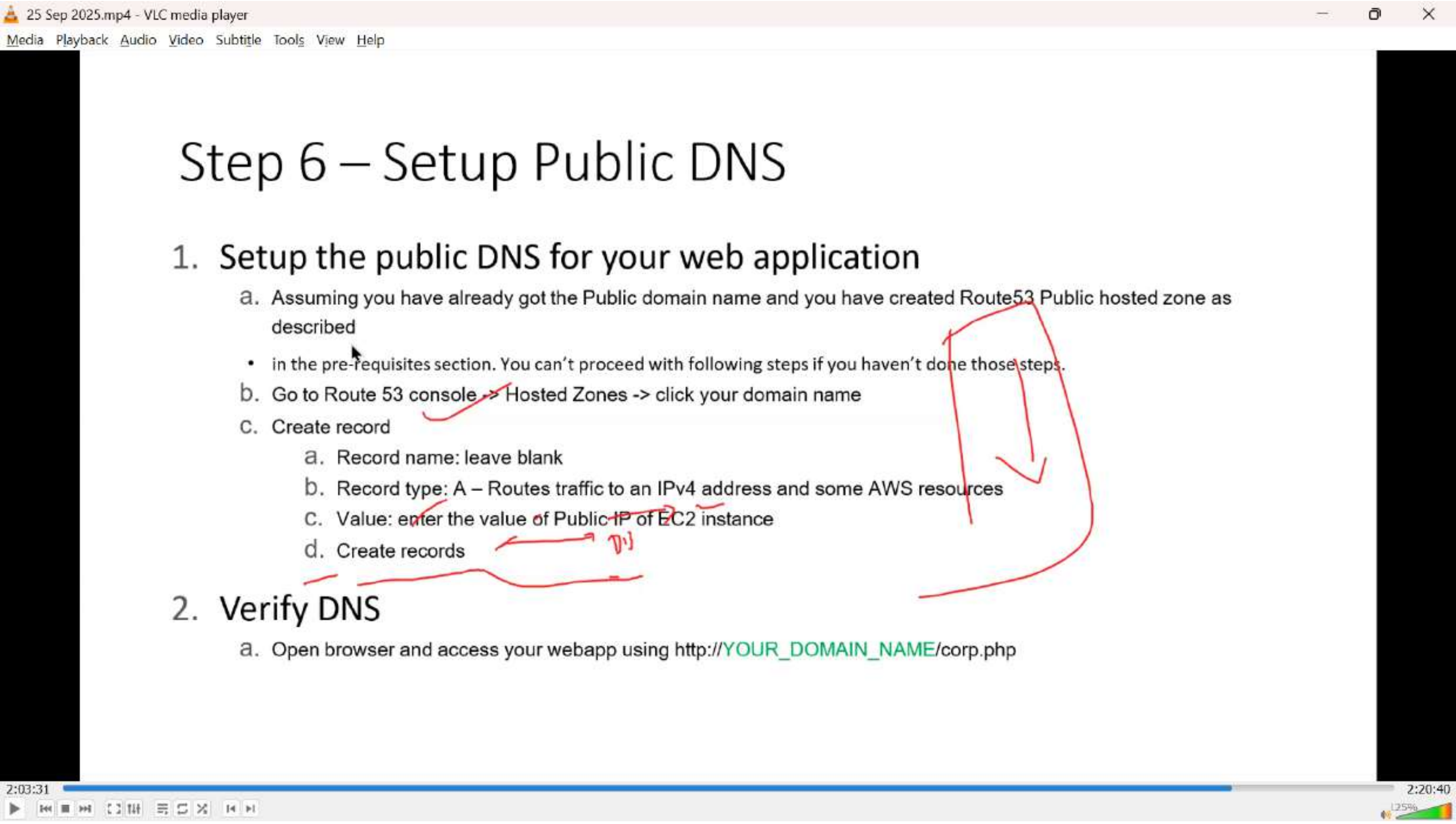
4. Create application file corp.php in /var/www/html directory.
   a. You can use the corp.php file from github repo:
5. Open your browser and hit the URL http://PUBLICIP/corp.php
6. Add few sample entries from the form on the webpage

Not secure | 43.205.213.38/corp.php

Welcome to my project website !

NAME	AGE	CITY	
			Add Data

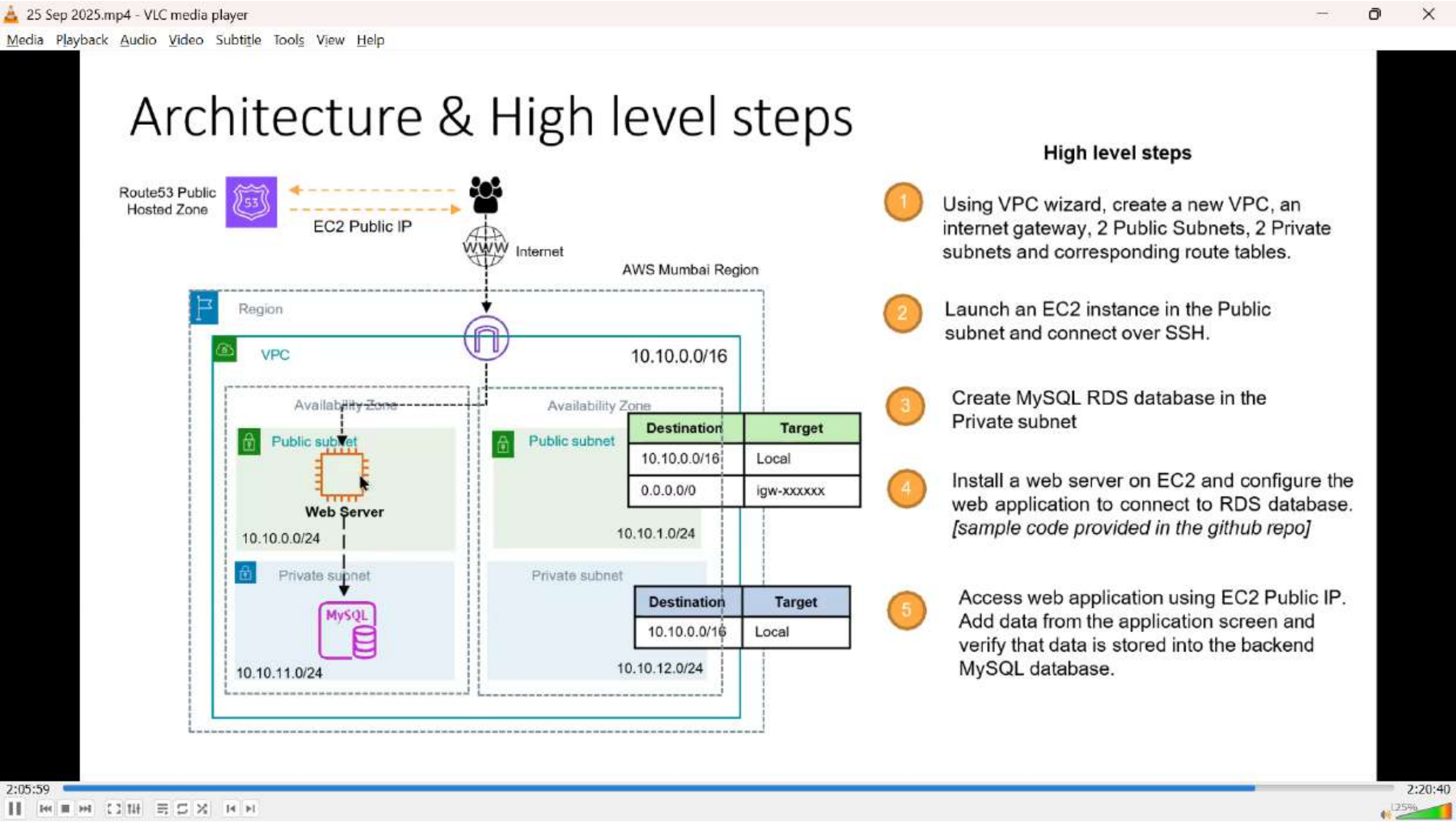
ID	NAME	AGE	CITY
1	Chetan	40	Pune



25 Sep 2025.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
2:03:31
2:20:40
125%

Step 6 – Setup Public DNS

1. Setup the public DNS for your web application
   a. Assuming you have already got the Public domain name and you have created Route53 Public hosted zone as described
   - in the pre-requisites section. You can't proceed with following steps if you haven't done those steps.
   b. Go to Route 53 console -> Hosted Zones -> click your domain name
   c. Create record
      a. Record name: leave blank
      b. Record type: A – Routes traffic to an IPv4 address and some AWS resources
      c. Value: enter the value of Public IP of EC2 instance
      d. Create records
2. Verify DNS
   a. Open browser and access your webapp using http://YOUR_DOMAIN_NAME/corp.php



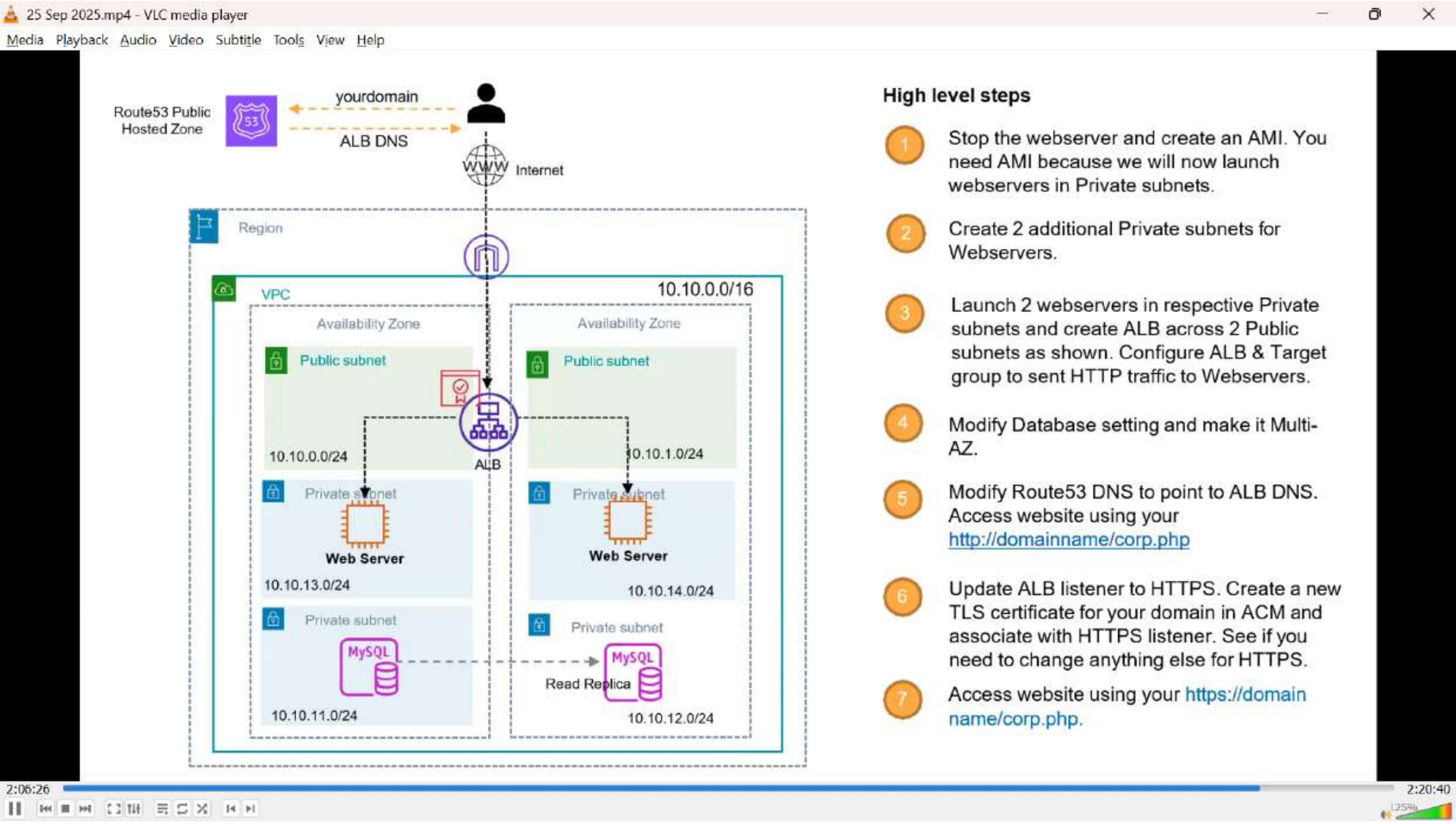
Architecture & High level steps

Destination	Target
10.10.0.0/16	Local
0.0.0.0/0	igw-xxxxxx

Destination	Target
10.10.0.0/16	Local

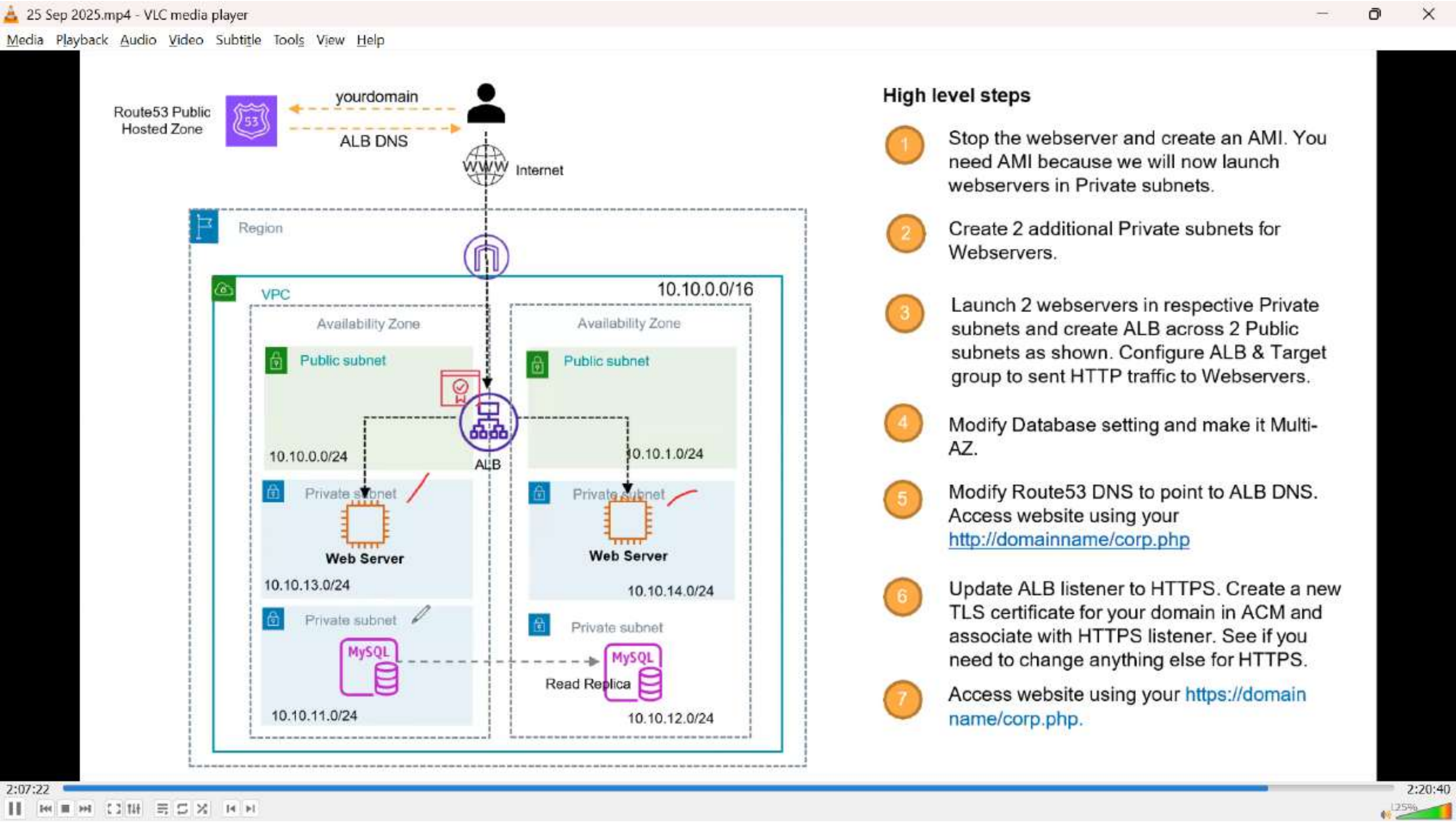
High level steps

1. Using VPC wizard, create a new VPC, an internet gateway, 2 Public Subnets, 2 Private subnets and corresponding route tables.
2. Launch an EC2 instance in the Public subnet and connect over SSH.
3. Create MySQL RDS database in the Private subnet
4. Install a web server on EC2 and configure the web application to connect to RDS database. *[sample code provided in the github repo]*
5. Access web application using EC2 Public IP. Add data from the application screen and verify that data is stored into the backend MySQL database.



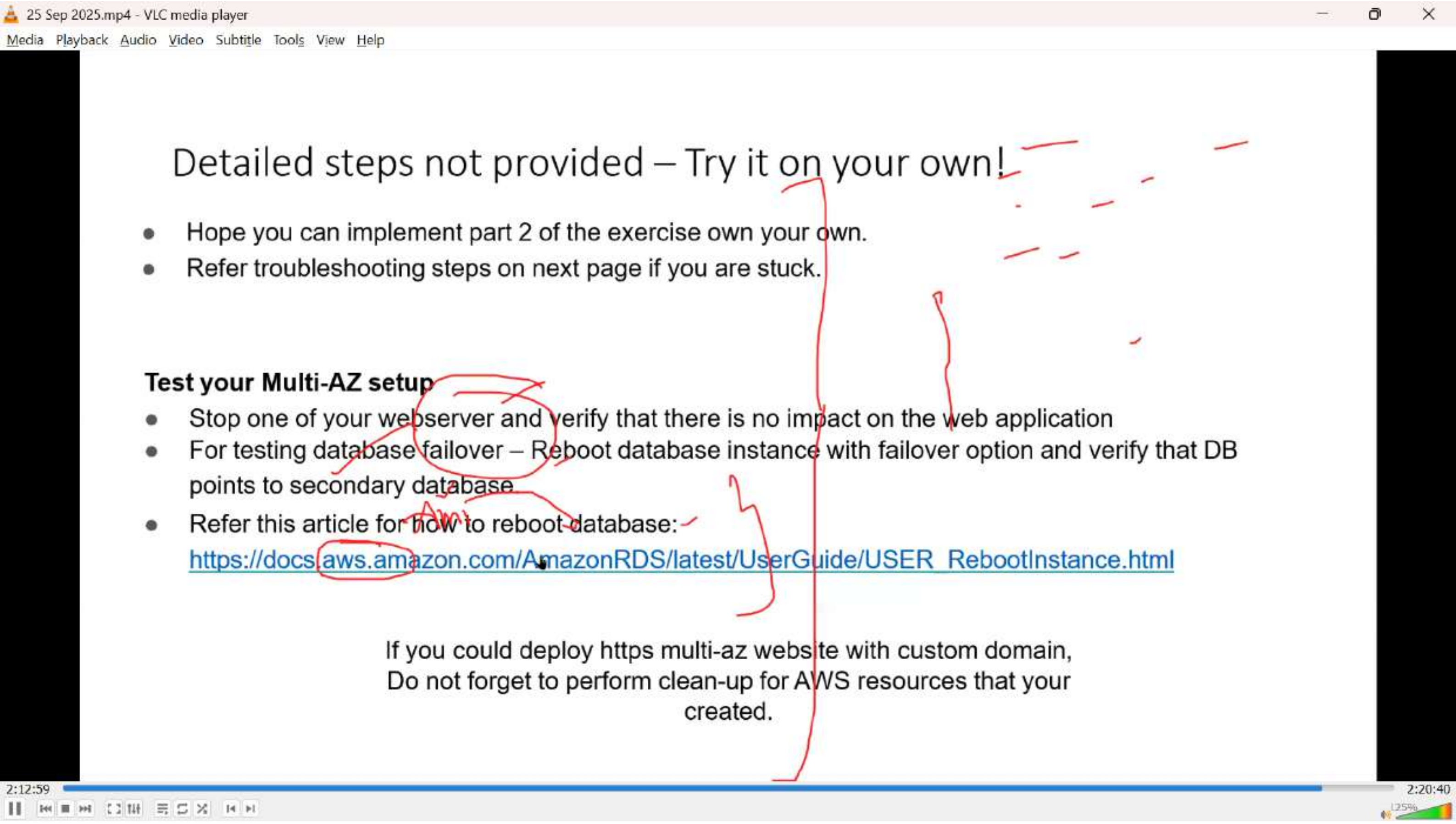
High level steps

1. Stop the webserver and create an AMI. You need AMI because we will now launch webservers in Private subnets.
2. Create 2 additional Private subnets for Webservers.
3. Launch 2 webservers in respective Private subnets and create ALB across 2 Public subnets as shown. Configure ALB & Target group to sent HTTP traffic to Webservers.
4. Modify Database setting and make it Multi-AZ.
5. Modify Route53 DNS to point to ALB DNS. Access website using your http://domainname/corp.php
6. Update ALB listener to HTTPS. Create a new TLS certificate for your domain in ACM and associate with HTTPS listener. See if you need to change anything else for HTTPS.
7. Access website using your https://domain name/corp.php.



High level steps

1. Stop the webserver and create an AMI. You need AMI because we will now launch webservers in Private subnets.
2. Create 2 additional Private subnets for Webservers.
3. Launch 2 webservers in respective Private subnets and create ALB across 2 Public subnets as shown. Configure ALB & Target group to sent HTTP traffic to Webservers.
4. Modify Database setting and make it Multi-AZ.
5. Modify Route53 DNS to point to ALB DNS. Access website using your http://domainname/corp.php
6. Update ALB listener to HTTPS. Create a new TLS certificate for your domain in ACM and associate with HTTPS listener. See if you need to change anything else for HTTPS.
7. Access website using your https://domain name/corp.php.


25 Sep 2025.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
2:12:59
2:20:40
125%

Detailed steps not provided – Try it on your own!

- Hope you can implement part 2 of the exercise own your own.
- Refer troubleshooting steps on next page if you are stuck.

Test your Multi-AZ setup

- Stop one of your webserver and verify that there is no impact on the web application
- For testing database failover – Reboot database instance with failover option and verify that DB points to secondary database
- Refer this article for how to reboot database: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_RebootInstance.html

If you could deploy https multi-az website with custom domain,
Do not forget to perform clean-up for AWS resources that your created.