

# AWS Networking Deep Dive: Elastic Load Balancing (ELB)

---

## INTRODUCTION TO THE ELB DEEP DIVE



**Ben Piper**

AWS CERTIFIED SOLUTIONS ARCHITECT

<https://benpiper.com>



# AWS Networking Deep Dive



Virtual Private Cloud



Elastic Load Balancing



Route 53 DNS



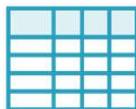
# You Must Know How to Create



VPCs



Subnets



Route tables



Security groups



# Prerequisites

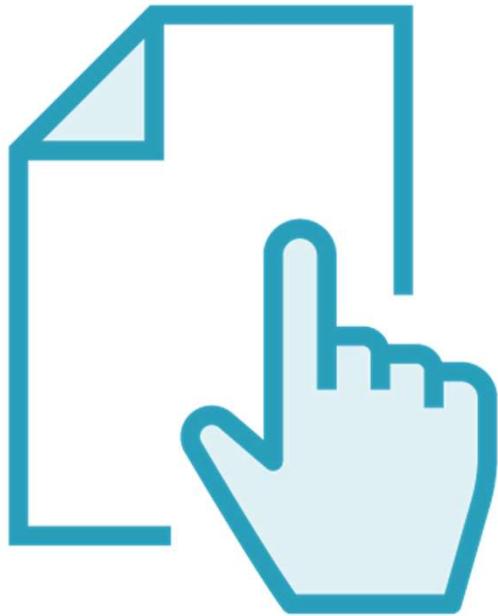


Create EC2 Linux instances  
using a particular AMI



SSH into those instances





AWS Networking Deep Dive: Virtual Private Cloud (VPC)

[netw.ink/aws-net-vpc](http://netw.ink/aws-net-vpc)



# Course Scenario

---



# Elastic Load Balancing

Improves fault tolerance and performance of applications running on EC2

Configuration specifics depend on application architecture



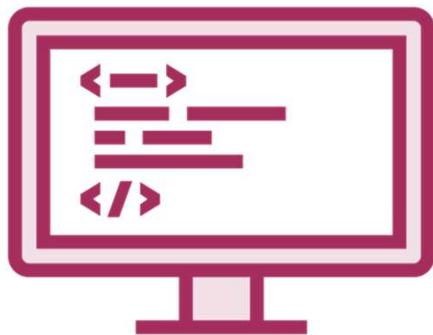
# Multi-tier Web Application

**Takes user input and stores it  
in a SQL database**

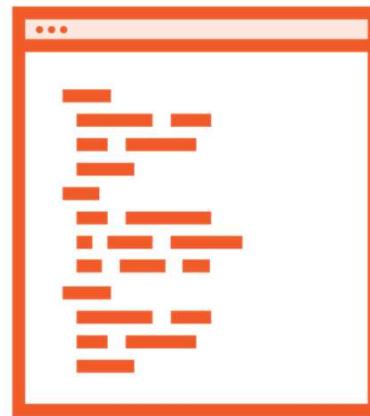
**Displays database contents on  
a webpage**



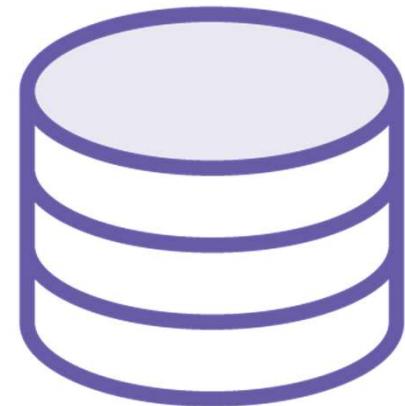
# Multi-tier Web Application



**Web tier**  
TCP/80 (HTTP)  
TCP/443 (HTTPS)

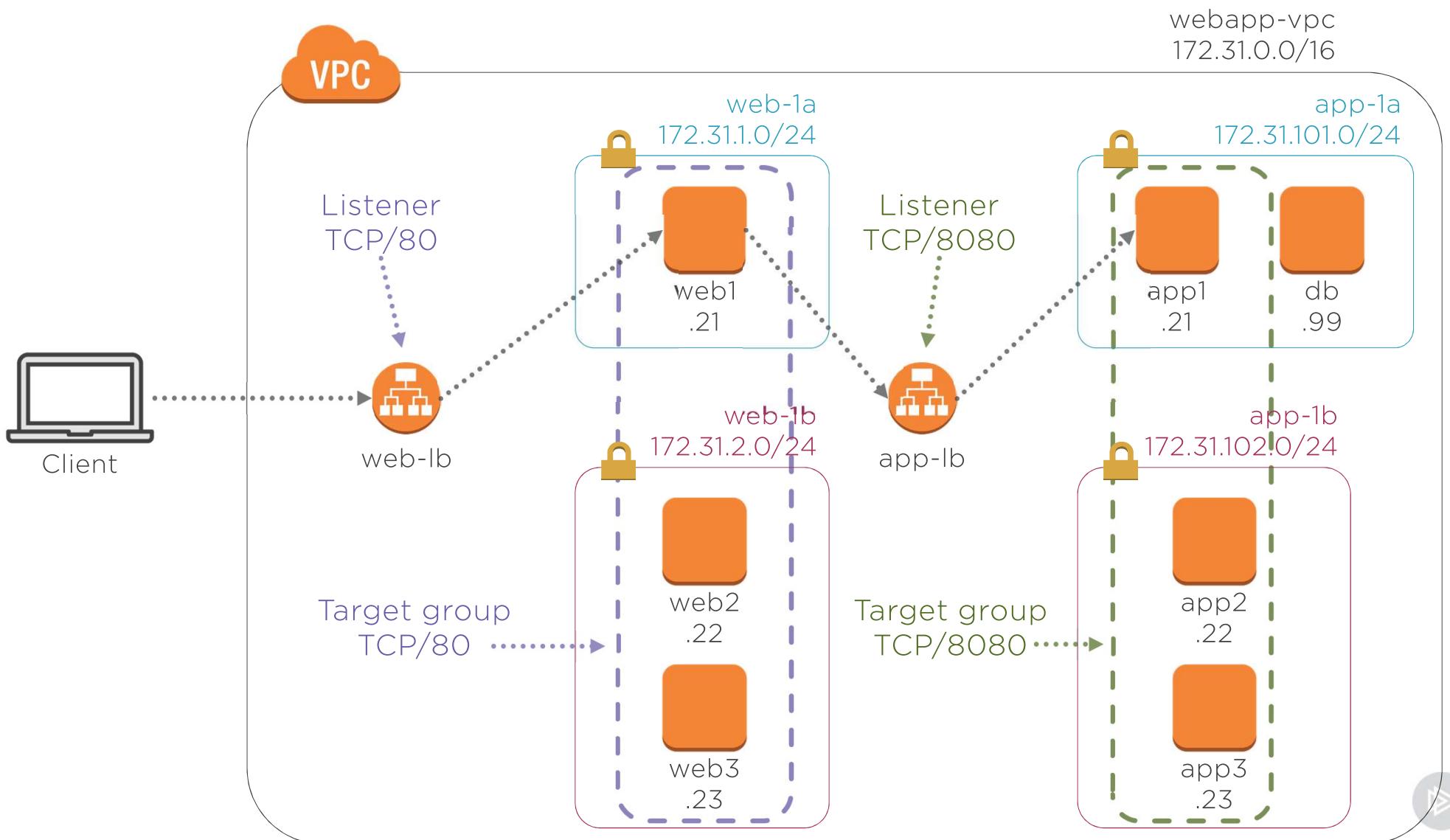


**Application tier**  
TCP/8080 (HTTP)  
TCP/8443 (HTTPS)



**Database tier**  
TCP/3306



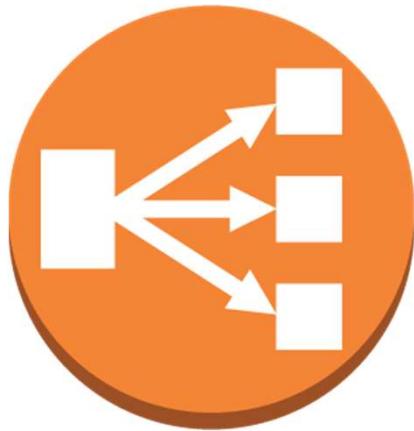


# Load Balancer Types

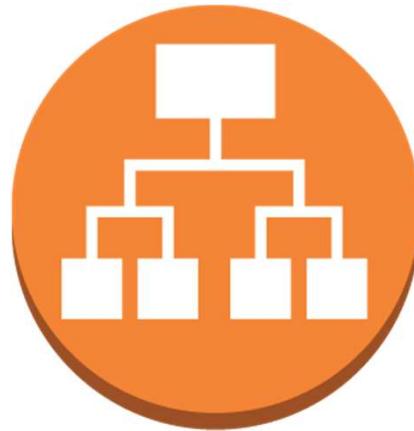
---



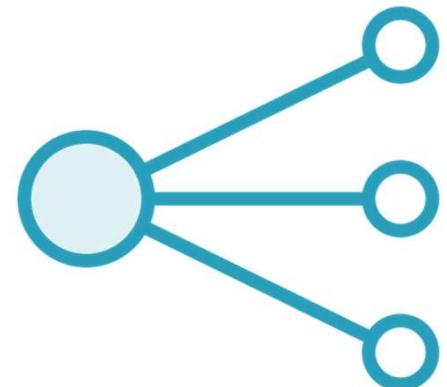
# Load Balancer Types



**Classic**  
(version 1)



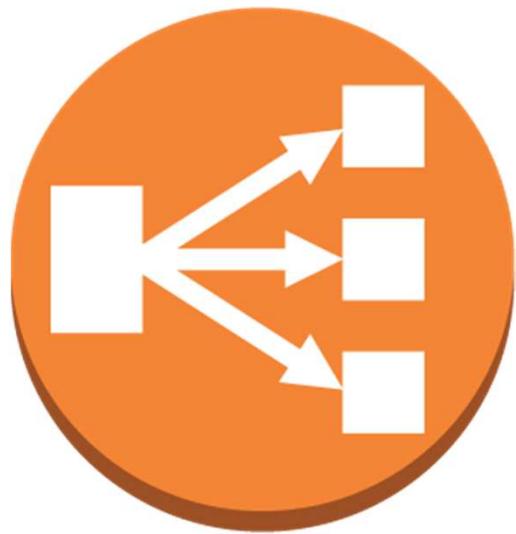
**Application**  
(version 2)



**Network**  
(version 2)



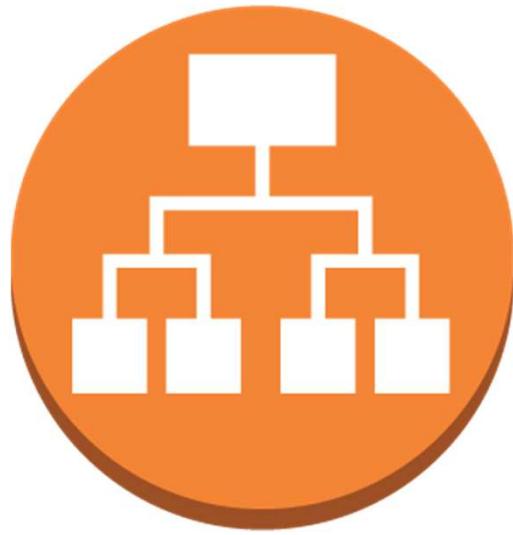
# Classic Load Balancer



**Designed for EC2-Classic network**  
**Not recommended for VPC**

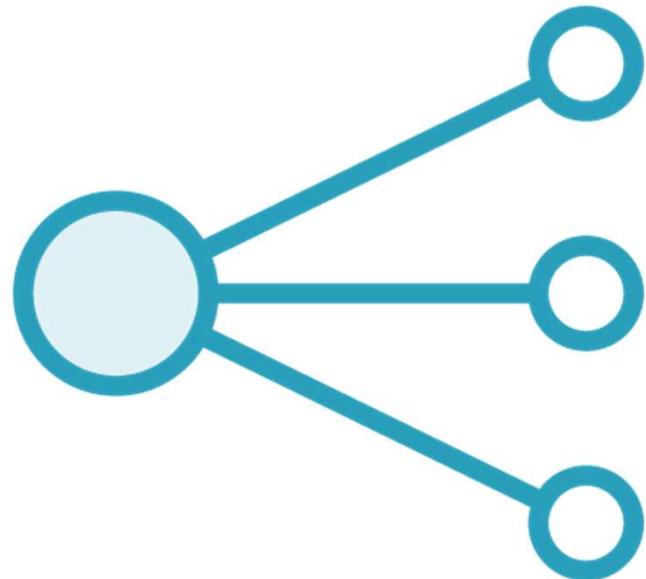


# Application Load Balancer



**HTTP and HTTPS traffic**  
**Terminates the client connection**  
**TCP ports 1-65535**  
**Listener supports IPv6**  
**Path and host-based routing**

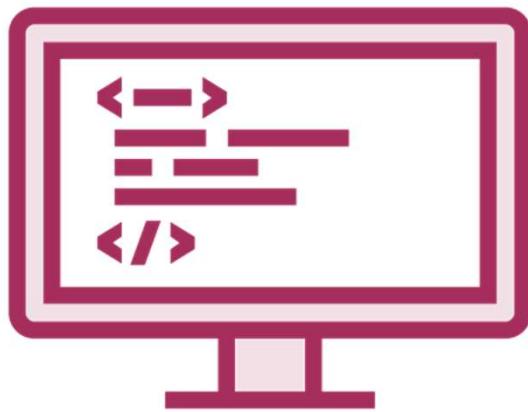
# Network Load Balancer



- Supports any TCP connection**
- Layer 4 load balancer**
- Handles high traffic at low latency**
- Does not terminate HTTP(S) connections**



# Application or Network Load Balancer?



**Web tier**  
TCP/80 (HTTP)  
TCP/443 (HTTPS)



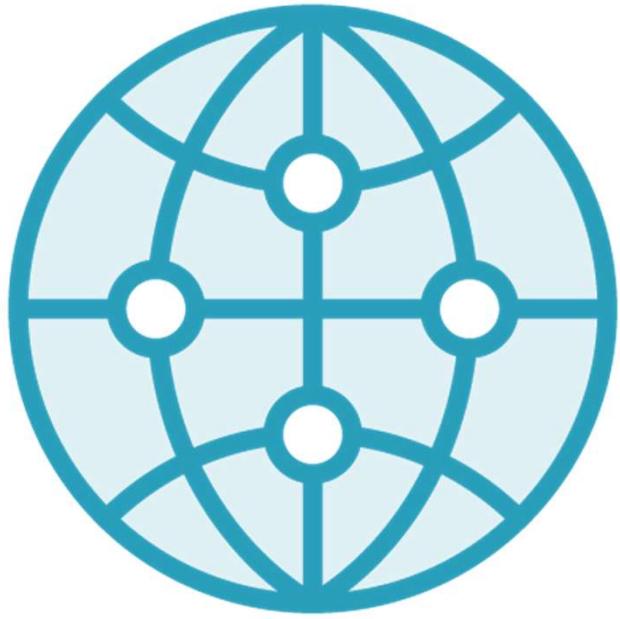
**Application tier**  
TCP/8080 (HTTP)  
TCP/8443 (HTTPS)



# Course Overview

---





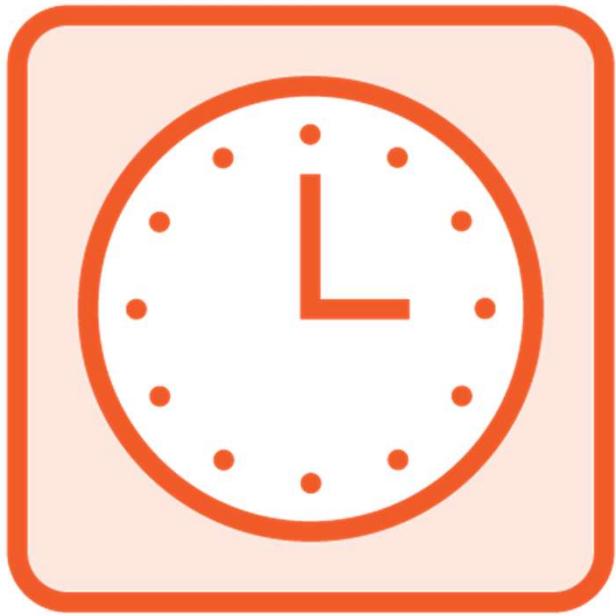
**Load Balancing Internet-facing  
Web Applications**





## Load Balancing Internal Web Services





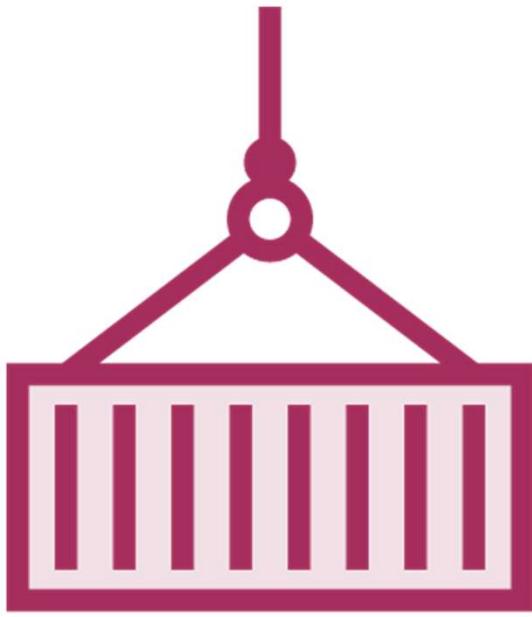
## Sticky Sessions and Idle Timeouts





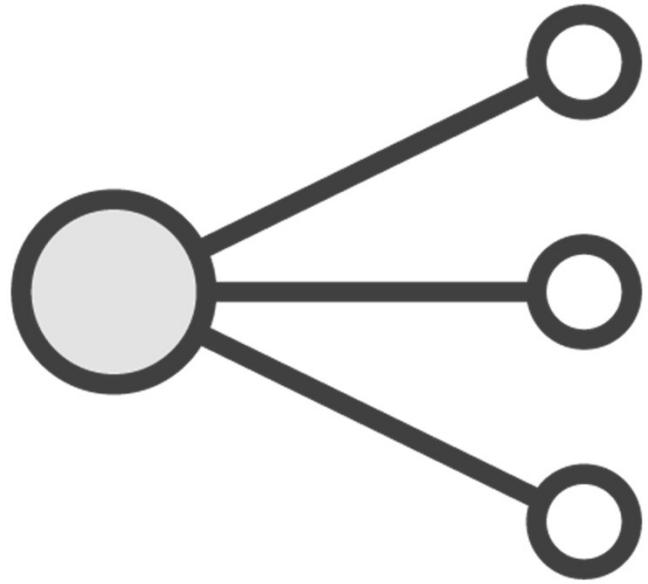
## Securing Web Applications with HTTPS





## Path-based Routing for Microservices





**Configuring the Network Load Balancer**



# Lab Setup

---



# VPC and Subnets

## **VPC: webapp-vpc**

- CIDR: 172.31.0.0/16
- Enable DNS hostnames
- Internet gateway: webapp-igw

## **Subnets:**

- web-1a: 172.31.1.0/24
- web-1b: 172.31.2.0/24
- app-1a: 172.31.101.0/24
- app-1b: 172.31.102.0/24



# Routes

## **Route table: webapp-rt**

- Associate with all subnets

## **Default routes:**

- IPv4: 0.0.0.0/0
- IPv6: ::0/0
- Target: webapp-igw



## Security Group: web-sg

Direction	Protocol	Ports	Source
Inbound	TCP	80	Any
Inbound	TCP	443	Any
Inbound	TCP	81	172.31.0.0/16
Inbound	TCP	22	Your IP



## Security Group: app-sg

Direction	Protocol	Ports	Source
Inbound	TCP	8080,8443	172.31.1.0/24
Inbound	TCP	8080,8443	172.31.2.0/24
Inbound	TCP	22	Your IP



## Security Group: db-sg

Direction	Protocol	Ports	Source
Inbound	TCP	3306	172.31.101.0/24
Inbound	TCP	3306	172.31.102.0/24
Inbound	TCP	22	Your IP



## Instance AMI

aws-elasticbeanstalk-amzn-  
2017.03.1.x86\_64-ecs-hvm-  
201709251832

ami-c710e7bd in the N. Virginia (US-East)  
region

If you use a different image, make sure  
Docker is installed

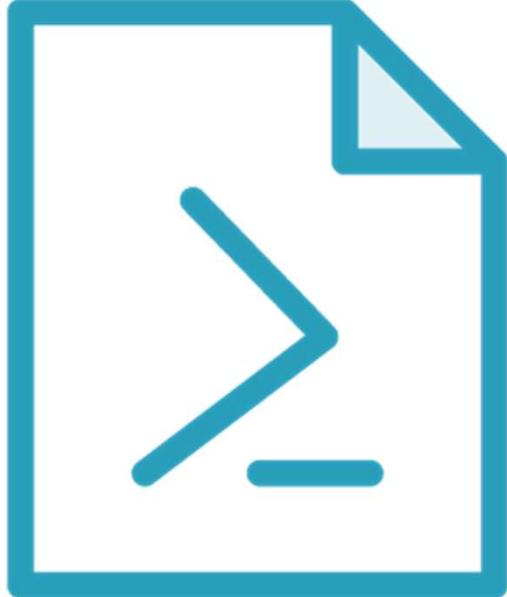


# Instances

Name	Subnet	IP address	Security group
web1	web-1a	172.31.1.21	web-sg
web2	web-1b	172.31.2.22	web-sg
web3	web-1b	172.31.2.23	web-sg
app1	app-1a	172.31.101.21	app-sg
app2	app-1b	172.31.102.22	app-sg
app3	app-1b	172.31.102.23	app-sg
db	app-1a	172.31.101.99	db-sg



# PowerShell Lab Setup Script



Available in the course exercise files and at  
[github.com/benpiper/aws-powershell](https://github.com/benpiper/aws-powershell)

Uses the AWS PowerShell SDK

Works on Windows, Linux, and Mac!

Refer to `e1b\lab-setup.md` for setup  
instructions



## Summary



**Understand what architecture, protocols, and ports your application uses**

**Application load balancer supports HTTP and HTTPS**

**Network load balancer supports any TCP-based connections**



## Coming Up Next



**Load Balancing Internet-facing  
Web Applications**

