

Automated Deployments with CloudFormation



Kien Bui

DevOps & Platform Engineer

Module Overview



CloudFormation fundamentals

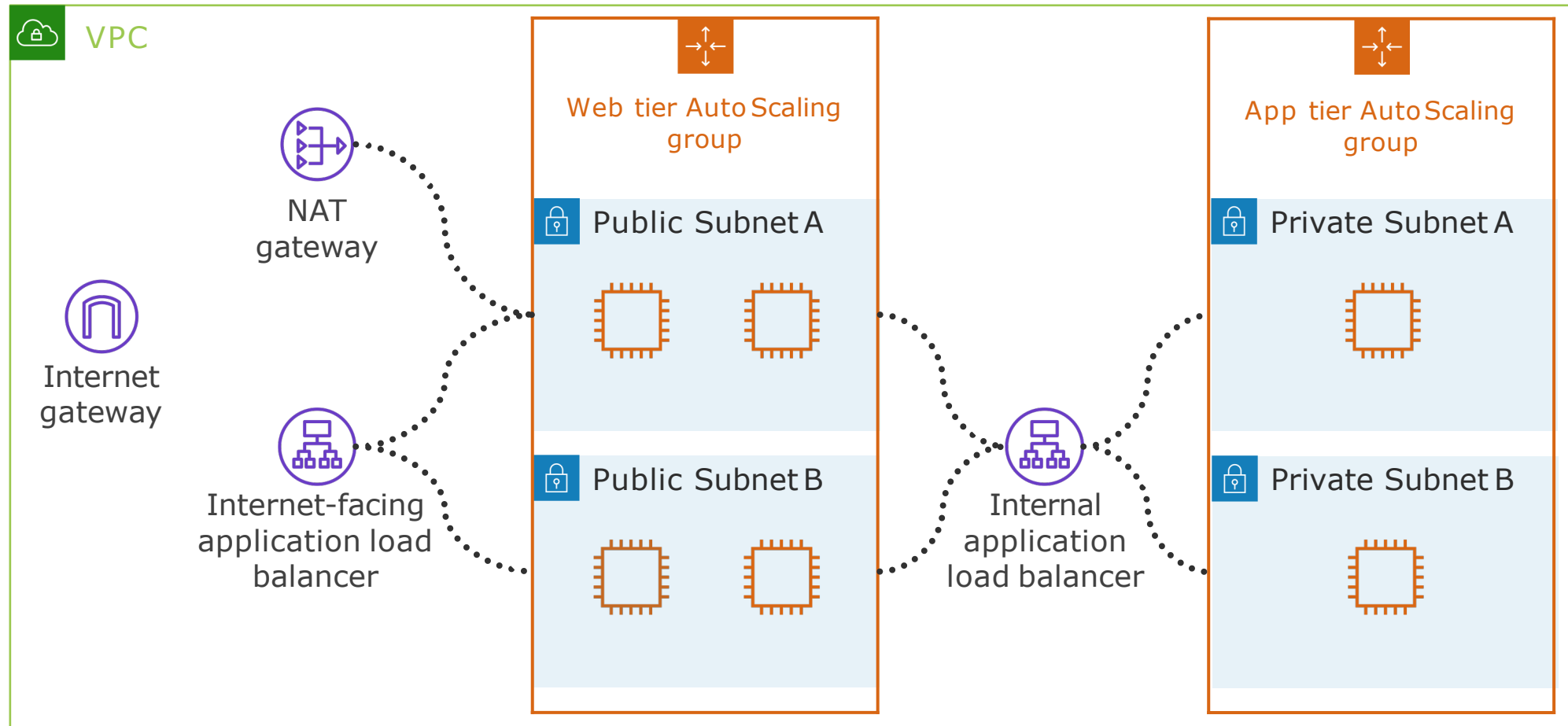
Review application architecture

Application load balancers

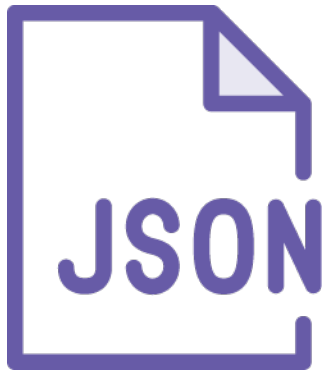
Auto Scaling groups

Deploy sample application

Sample Application Architecture



Template



JSON or YAML document that describes
AWS resources

Infrastructure as code

Used to create a stack

Stack



Created by a template

Collection of resources that you create, update, and delete as a single unit

You can manually manage individual resources in a stack

Multiple Templates



Different teams manage different resources

Resources have different lifecycles

Distributing resources across different stacks makes them easier to manage

CloudFormation Templates

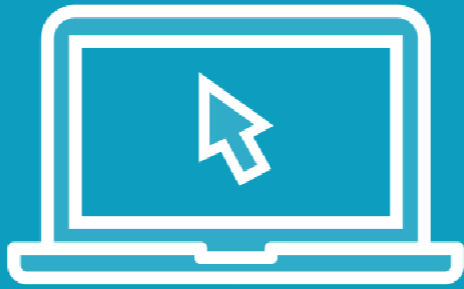


Download them from
https://TOOA01@dev.azure.com/TOOA01/DevOps%20Basic%20Online/_git/aws-fun-deep-01

- app-stack.json
- network-stack.json

Reviewing the CloudFormation Templates

Demo



Load `app-stack.json` and `network-stack.json` into your favorite text editor

Nested Stack



app-stack.json creates the parent stack

It calls network-stack.json to create the nested stack

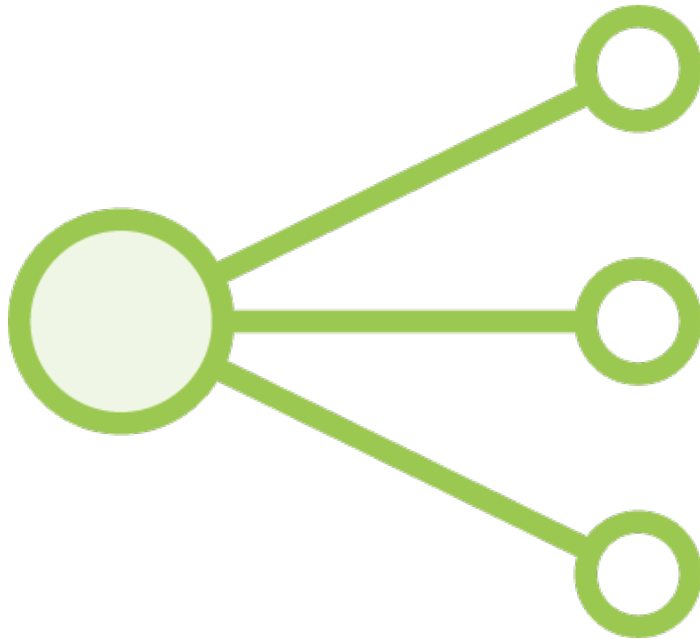
Stack Output



Key-value pair that CloudFormation makes available to other stacks and via the `aws cloudformation describe-stacks` CLI command

Application Load Balancers

Application Load Balancer (ALB)



Supports HTTP and HTTPS traffic

You can use any TCP port

Listener receives connection from a client and proxies it to an instance in the target group

Uses round-robin load balancing by default

Can monitor health of instances

Application Load Balancer Schemes

Internet-facing

Reachable from the internet

Public IP addresses

Public DNS name

Internal

Not reachable from the internet

Private IP addresses

Private DNS name

Health Checks



Each instance must pass its health check before receiving traffic

Sends HTTP GET request and looks for a success code

HTTP Request and Response

Protocol: HTTP

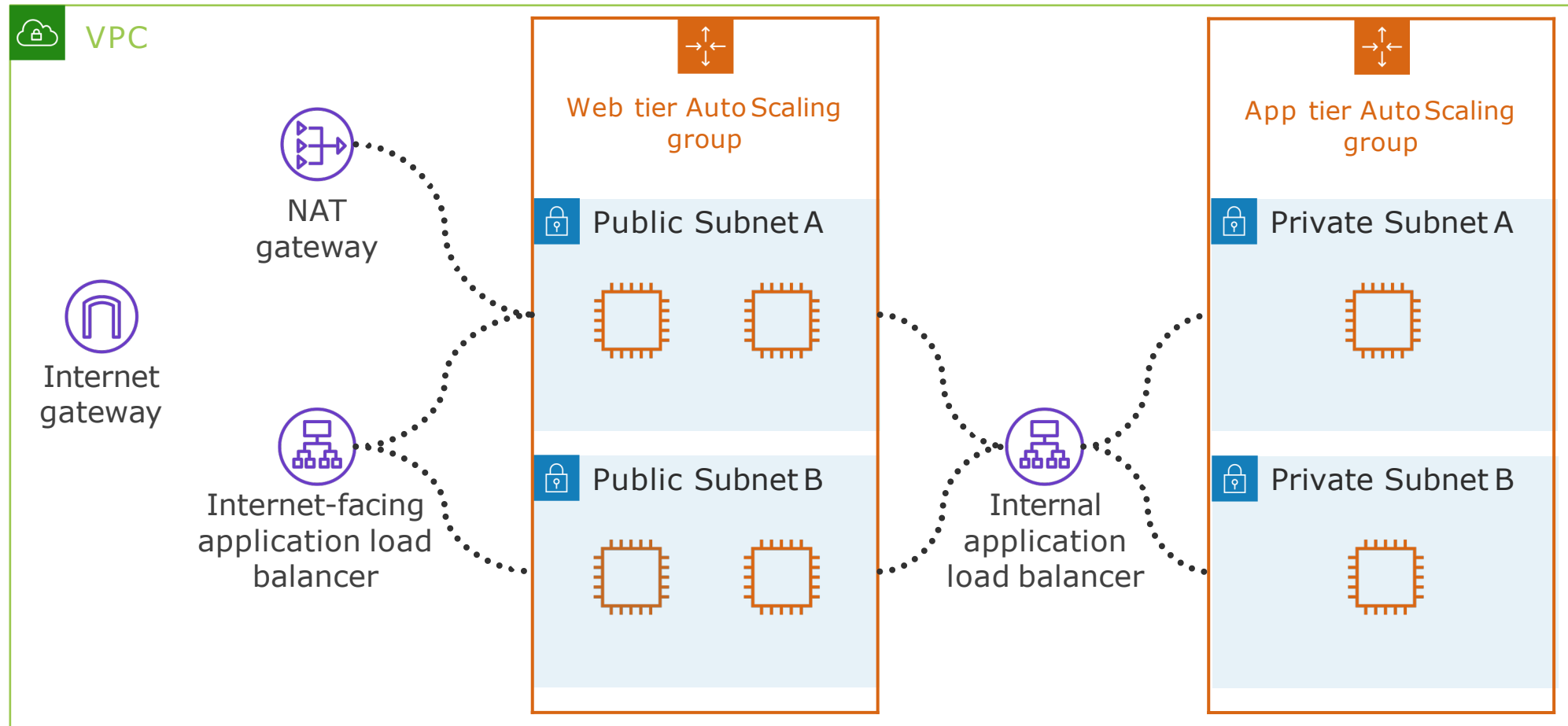
Port: TCP/80

Request Method: GET

Status Code: 200 OK

Auto Scaling Groups

Sample Application Architecture



Auto Scaling Tasks



Launch a certain number of instances into the Auto Scaling group

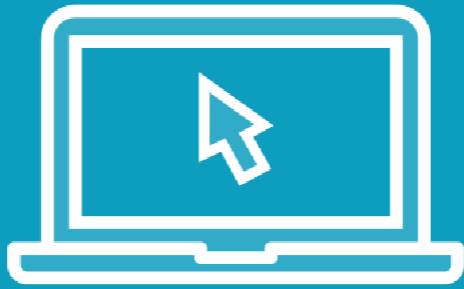
Add the instances to the ALB target group

Terminate and recreate unhealthy instances

Scale in or out based on average group CPU utilization

Deploying the Stack

Demo



Deploy the application stack from app-stack.json

Refer to the aws-cli-commands.txt file

Summary

Summary



CloudFormation saves time and money and reduces errors

- Fast deployments
- Stacks are all-or-nothing

Summary



Elastic load balancing and Auto Scaling work together

- ELB provides health checks
- Auto Scaling adds instances to the ELB target group

Coming up Next



Multi-region applications with Route 53