Automating AWS with CloudFormation

INTRODUCTION

Manage Infrastructure Manually

- Good way to start
- Perfect for Proof-of-Concept and Prototyping

Automate Infrastructure

- Improve quality and efficiency
- Major benefit of using AWS

Background Knowledge

Virtual Machine	Private Network	Object Storage	NoSQL Database
EC2	VPC	S3	DDB

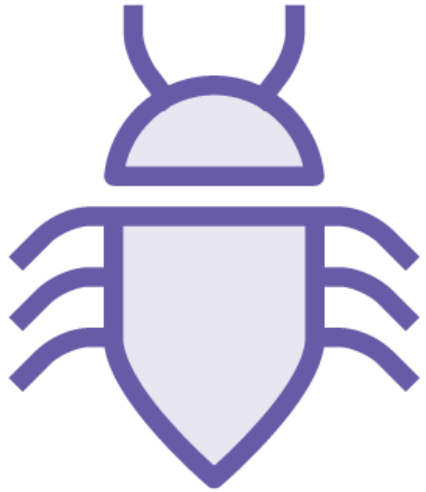
Security Group	SQL Database	Route53 DNS	
SG	RDS	R53	...

Automation vs. Manual Work

Managing AWS Manually

High costs

Managing infrastructure manually is inefficient



Low quality

Failures and outages occur during manual changes

Low flexibility

Changing infrastructure is complex and therefore avoided

Adam, DevOps engineer at Cotocus

Frustrated with current situation

Goals

- Lower costs
- Improve quality
- Improve flexibility

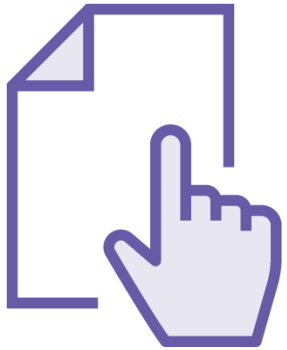
Advantages of Automating Infrastructure

Speed up

Replacing manual work improves efficiency

Automated testing

Testing infrastructure changes like testing software



Documentation

Code documents how to manage infrastructure

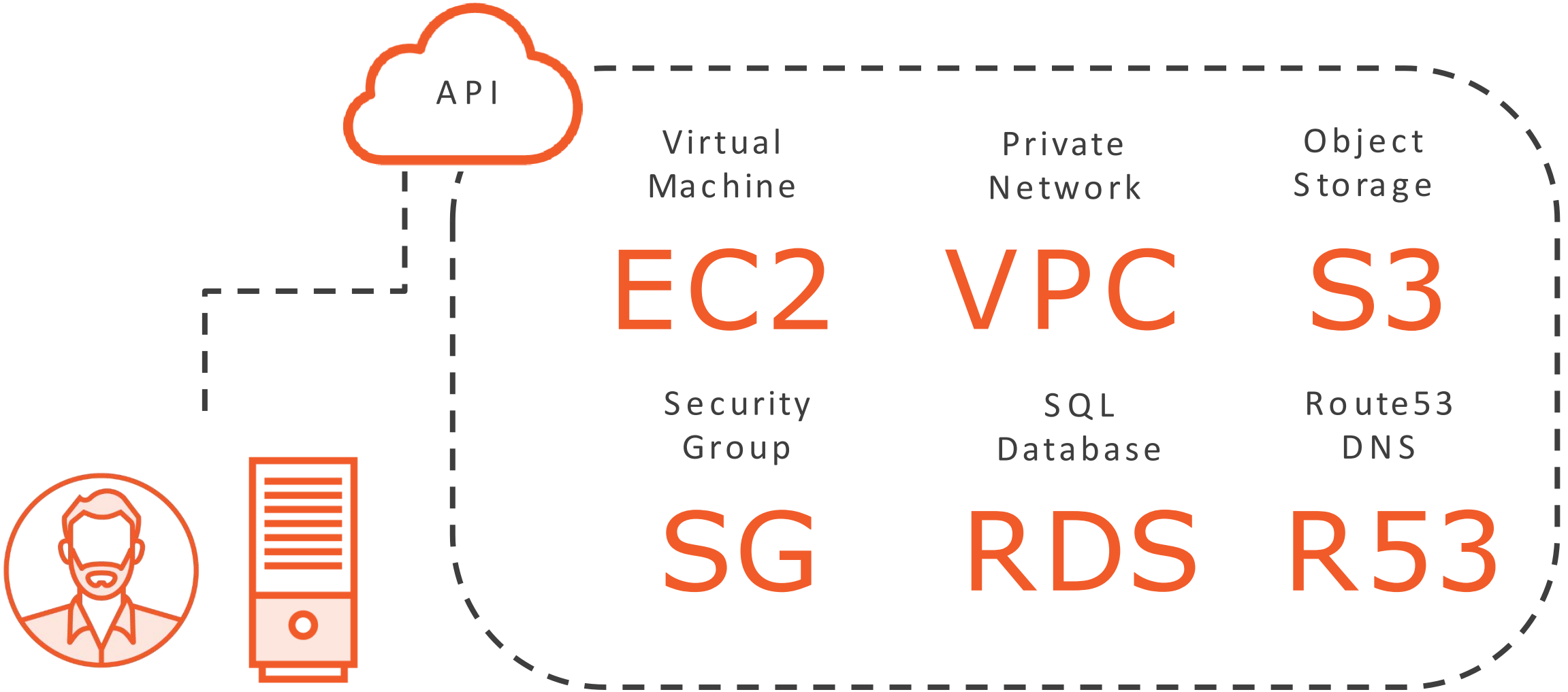
What is the biggest advantage of Amazon Web Services?

Every part of AWS is controllable via an

API

AWS API Enables Automation

Controlling AWS with an API



API Usage Examples

Launch EC2 instance

Create Security Group

Update RDS database configuration

Create networking infrastructure

Add new IAM user

…

Tools to Access AWS API

Command Line Interface (CLI)

Access API from Terminal or write small Shell scripts

Software Development Kit (SDK)

Access API from programming language of your choice: Java, PHP, JavasScript, Ruby, Python, …

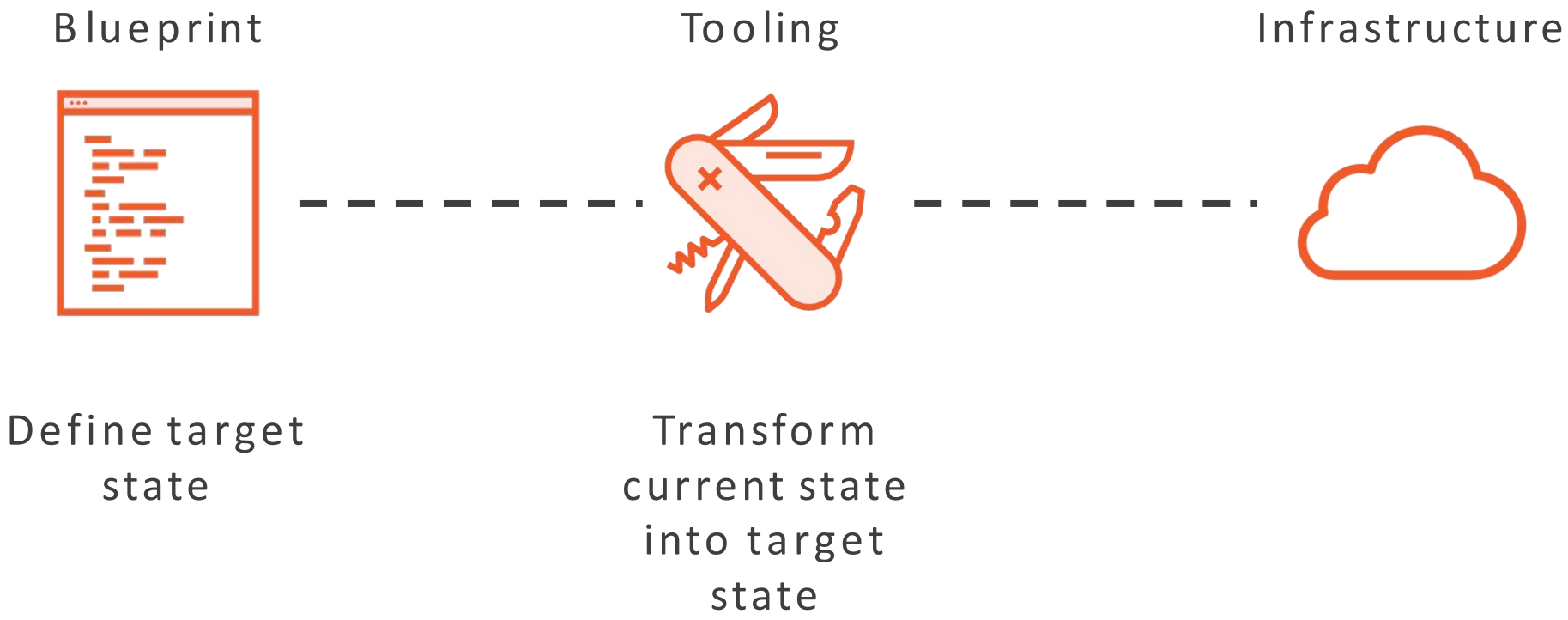
Resolve dependencies

- RDS database needs to be created before EC2 instances are created.
- Security Groups needs to be created before EC2 instance is created.

Update infrastructure

- Which steps are necessary to update existing infrastructure?
- Hard coded transformation not very flexible.

Describing Infrastructure in Code



Describing infrastructure in code

- Resolves dependencies automatically
- Creates, updates or deletes infrastructure
- Performs idempotent and predictable changes to infrastructure

Describing AWS infrastructure in code

Supports 95% of services on AWS

Maintained by Amazon

Free to use

AWS CloudFormation

CloudFormation Key Concepts

Features of CloudFormation

Create

Creating resources in the right order based on their dependencies.

Update

Updating existing stacks by making changes to existing resources.

Delete

Deleting all resources of a stack in the right order based on their dependencies.

Automate almost every part of AWS with CloudFormation

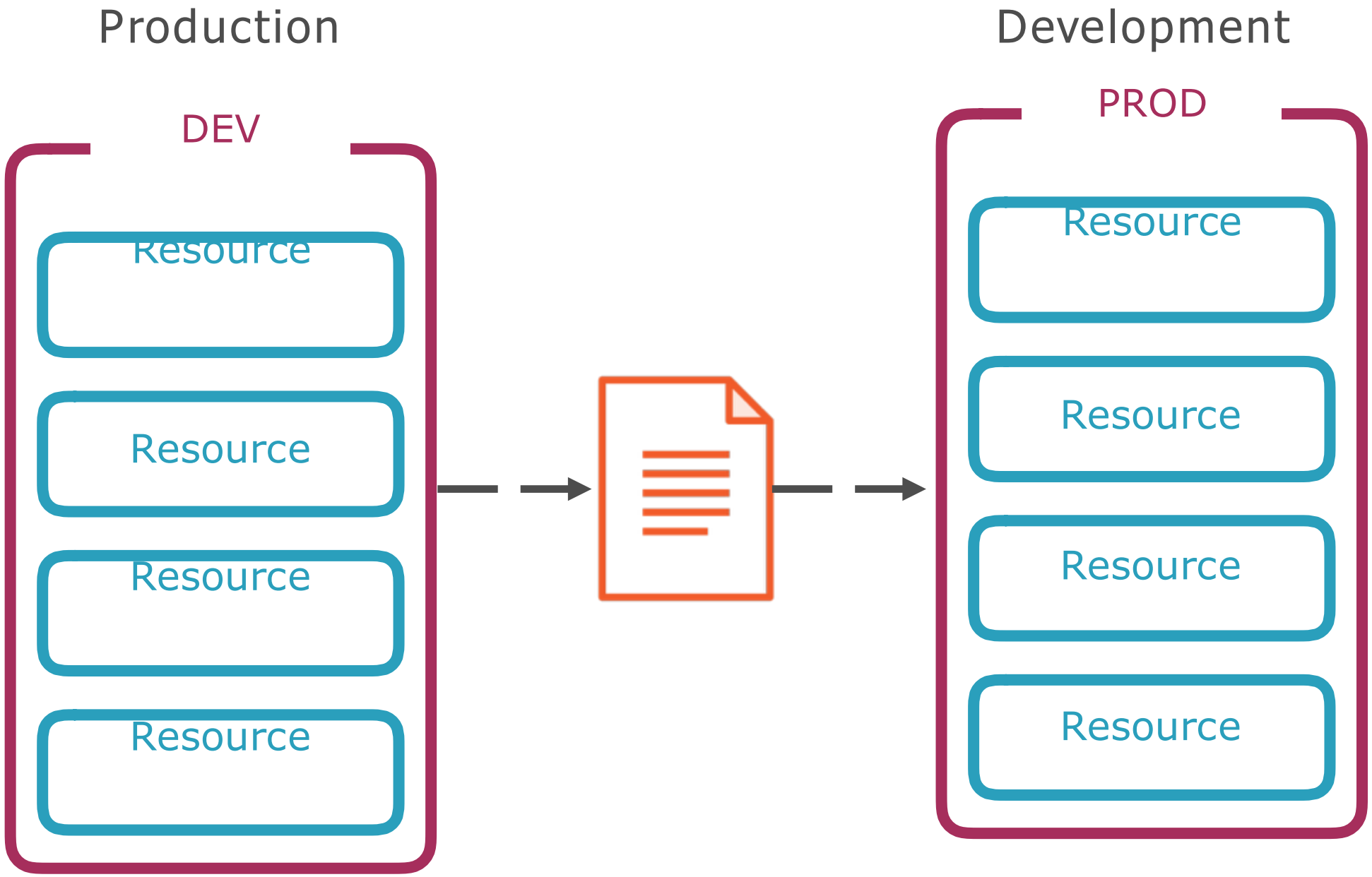
Learn how to automate:

- EC2 Instance
- VPC
- Security Group
- RDS

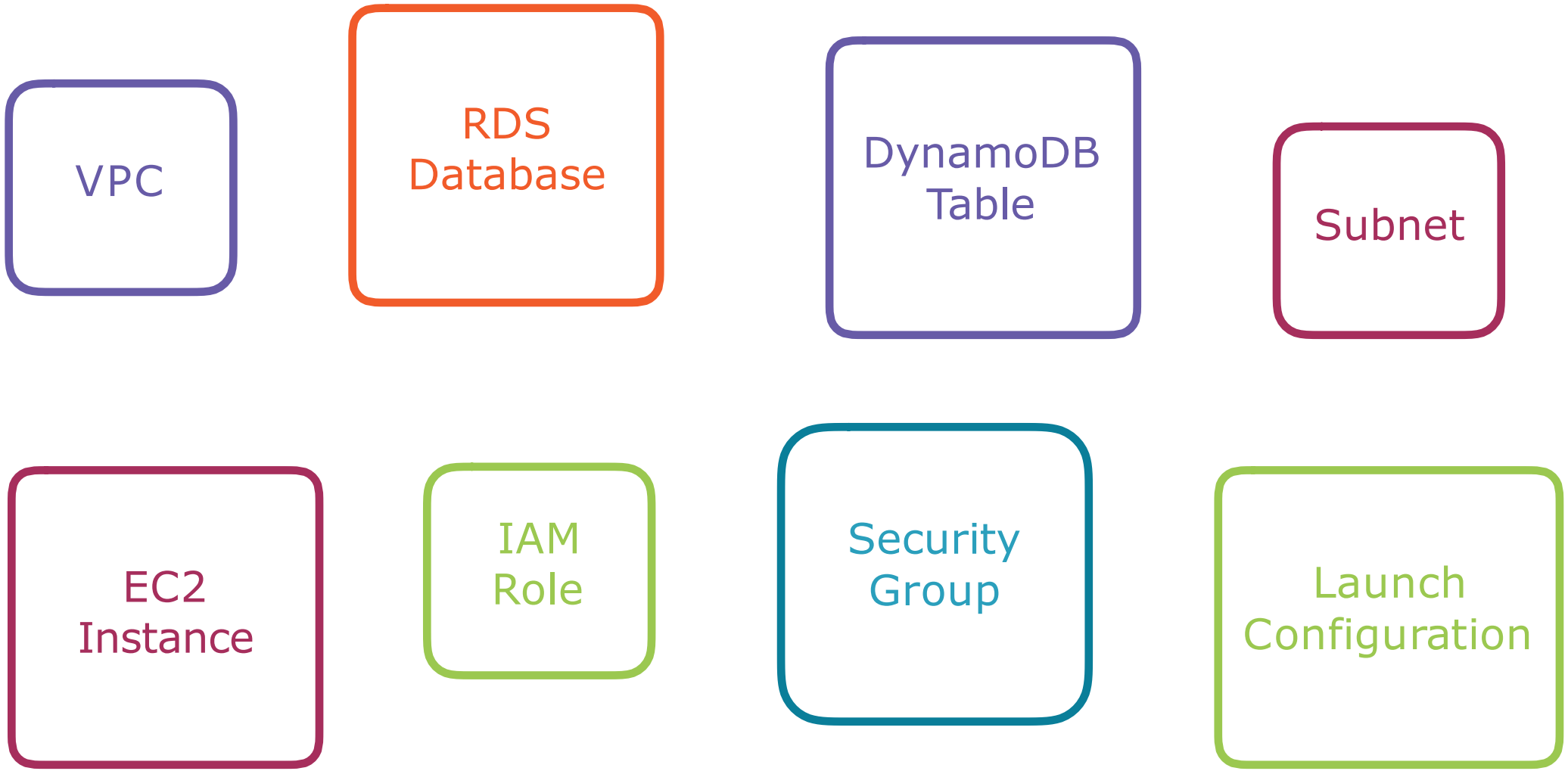
Use CloudFormation to create stacks

- Managment Console
- Command Line Interface

Create CloudFormation templates


Production
Development
DEV
PROD
Resource
Resource
Resource
Resource
Resource
Resource
Resource
Resource

Using Cloud Formation, You may



Conclusion

Automating infrastructure

- Lower costs
- Improve quality
- Improve flexibility

Describing infrastructure in code

- No scripting or programming needed
- Perfect tool for the job

AWS CloudFormation

- Describing AWS infrastructure in code
- Highly integrated with AWS

Let’s start!

Objective

Create, Test, and Delete Infrastructure with CloudFormation

CloudFormation Templates

Service to provision resources using templates

CloudFormation Template

JSON Document

Contains configuration for resources Can be used in

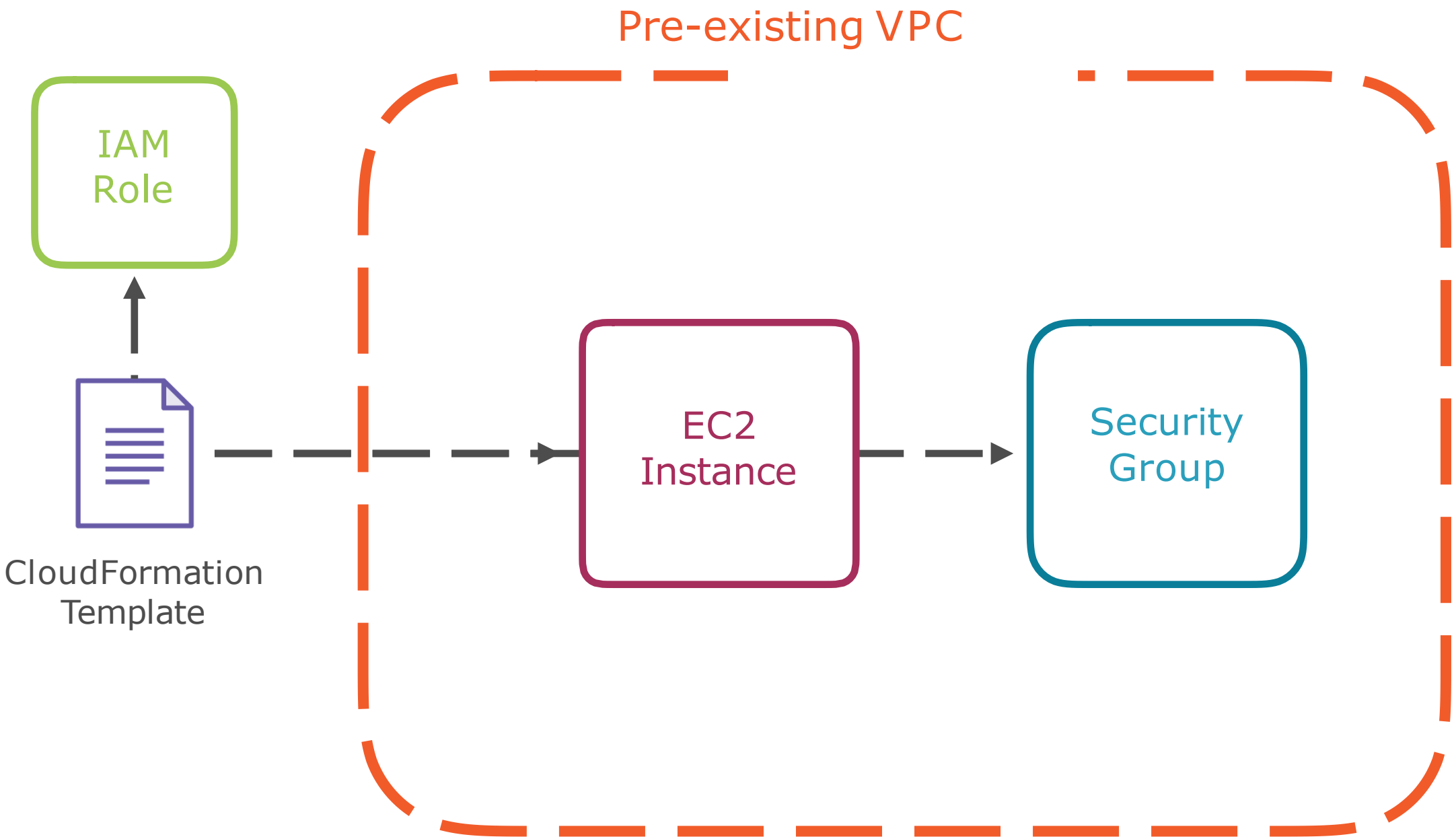
Version Control

No limit to amount of resources

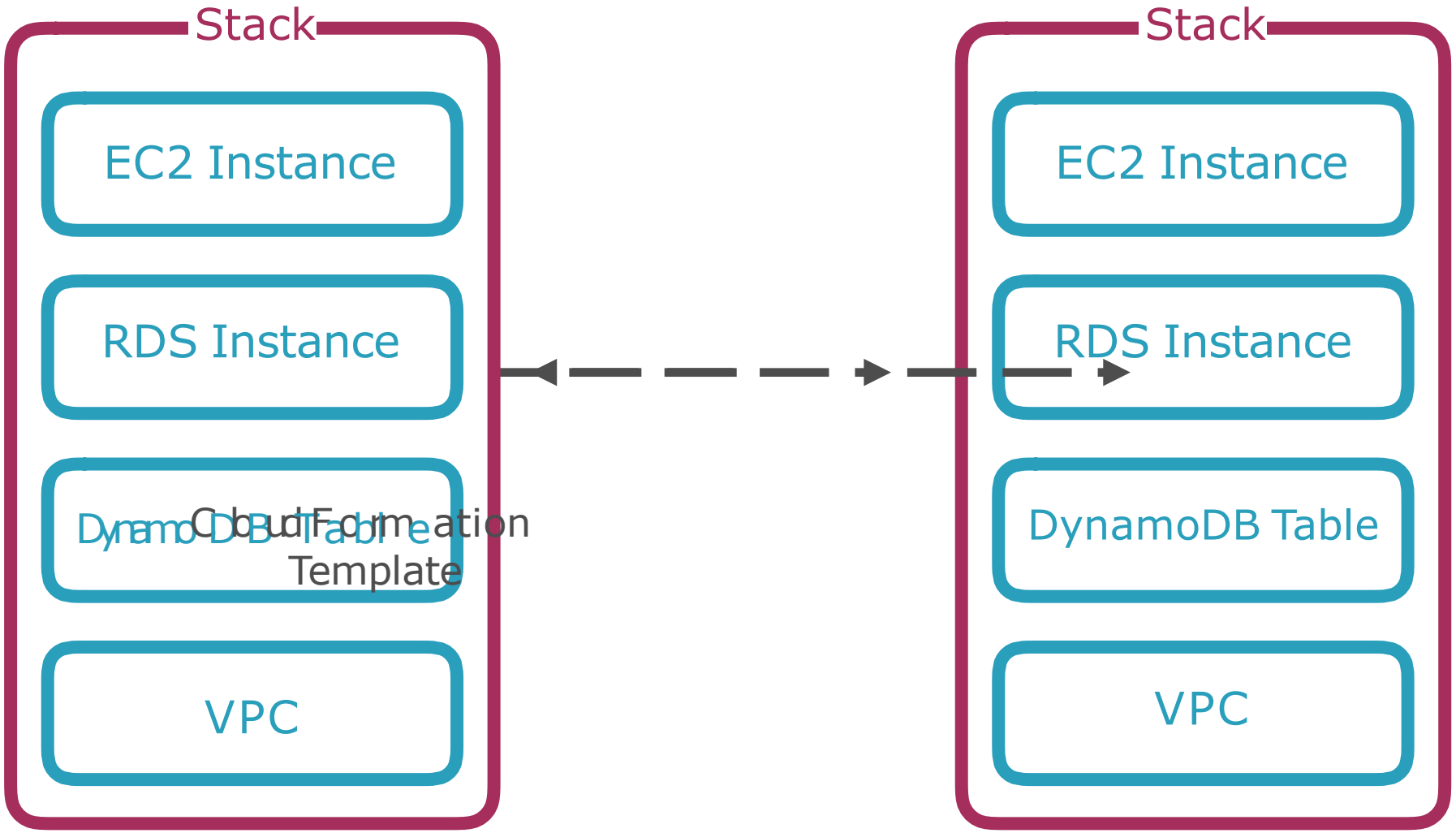
```
"Type": "AWS::EC2::Instance",
"Properties": {
    "ImageId": "ami-bff32ccc",
    "InstanceType": "t2.nano”
}
```

Template

Written in JSON describing the target state of the infrastructure.


Pre-existing VPC
IAM Role
CloudFormation Template
EC2 Instance
Security Group

CloudFormation Stack



Other CloudFormation Operations

Updating a Stack

Deleting a Stack

Updating a stack may require resources to restart

Deleting a stack removes all resources created in that stack

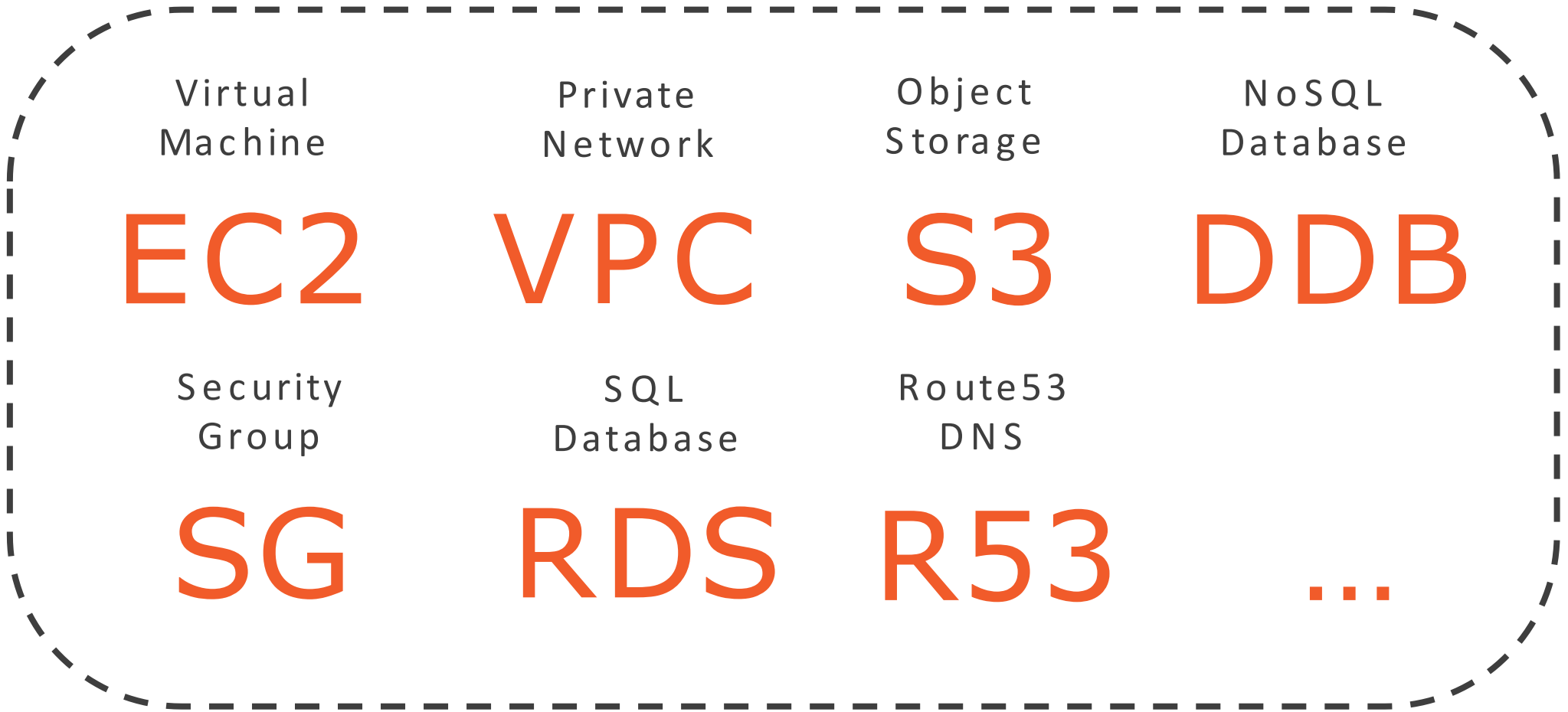
CloudFormation only creates the resources you explicitly configure in the template

CloudFormer

Creates a CloudFormation template based on existing infrastructure

1Template
=
Multiple Stacks

Stack, Instance of a Template



AWS CloudFormation

JSON-Template

Define all needed AWS resources and configure them

CloudFormation

CF

Transform current state into target state

Stack

Instance of template running in an AWS region

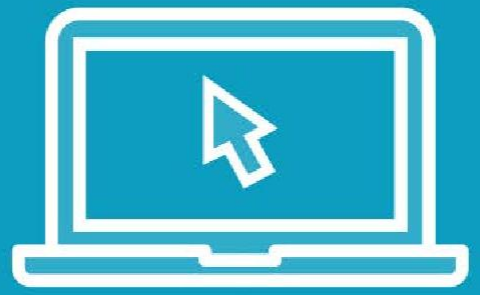
Custom Resources

Custom Resources

Lambda-backed custom resources

SNS-backed custom resources

Demo



Simple WordPress environment

- EC2 instance running Web Server
- RDS instance with MySQL engine
- Security Groups

Have a look at template defining target state

Create stack based on template

Text Editor

IDE Integration

- Eclipse
- Visual Studio

AWS CloudFormation Designer

- Graphical tool for creating templates
- Part of AWS Management Console