AWS Architecting and SysOps

Monitoring and Deploying AWS Resources, Part 2

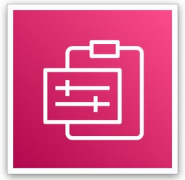
June-July 2019

Contents

Amazon Lightsail

Cheap VPS service on AWS

Provisioning & Deployment



AWS Elastic Beanstalk

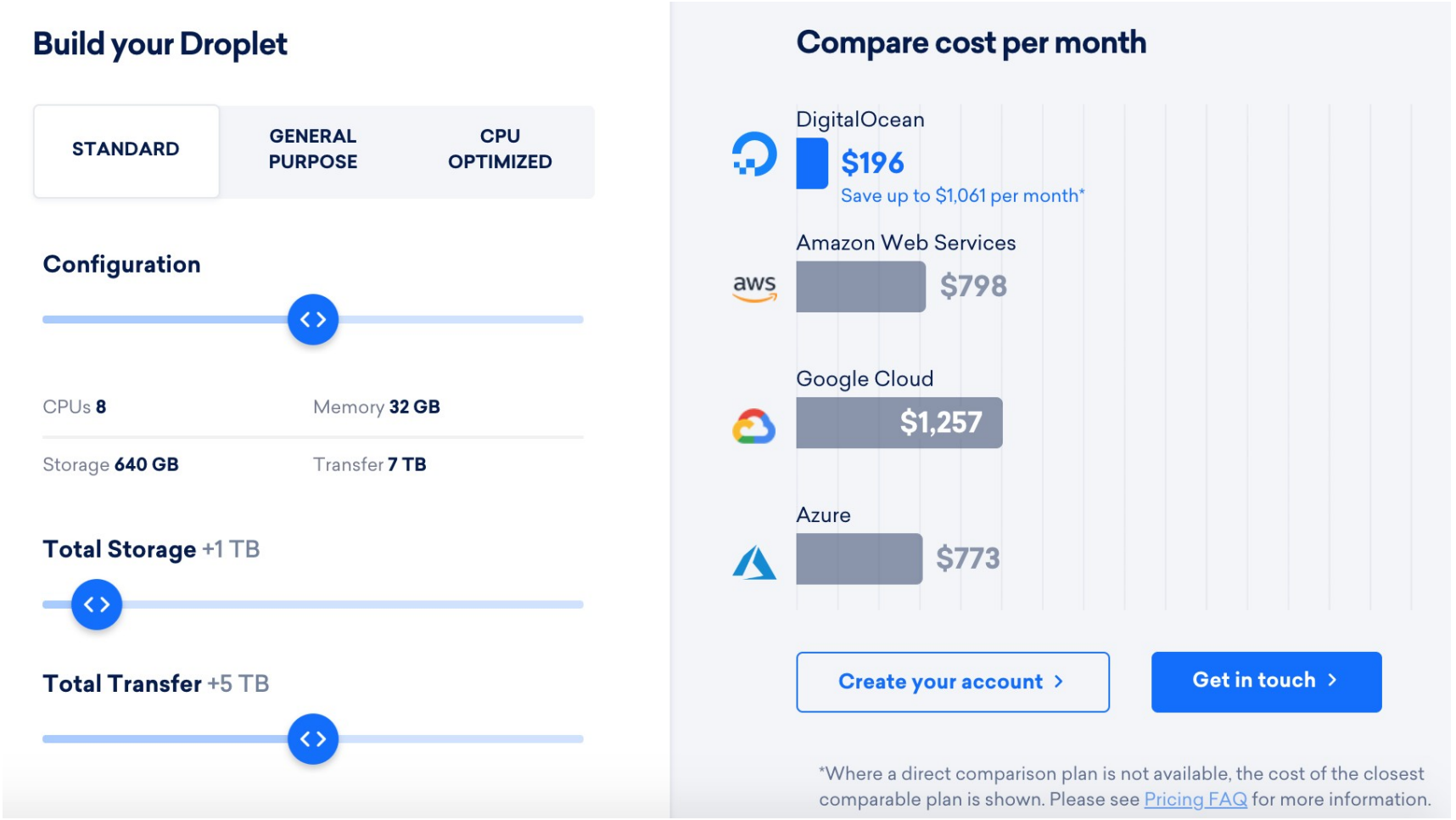
AWS CloudFormation

Amazon Lightsail

AWS OpsWorks

Why Amazon Web Services?

- There are several, very cheap services for hosting servers on the Cloud



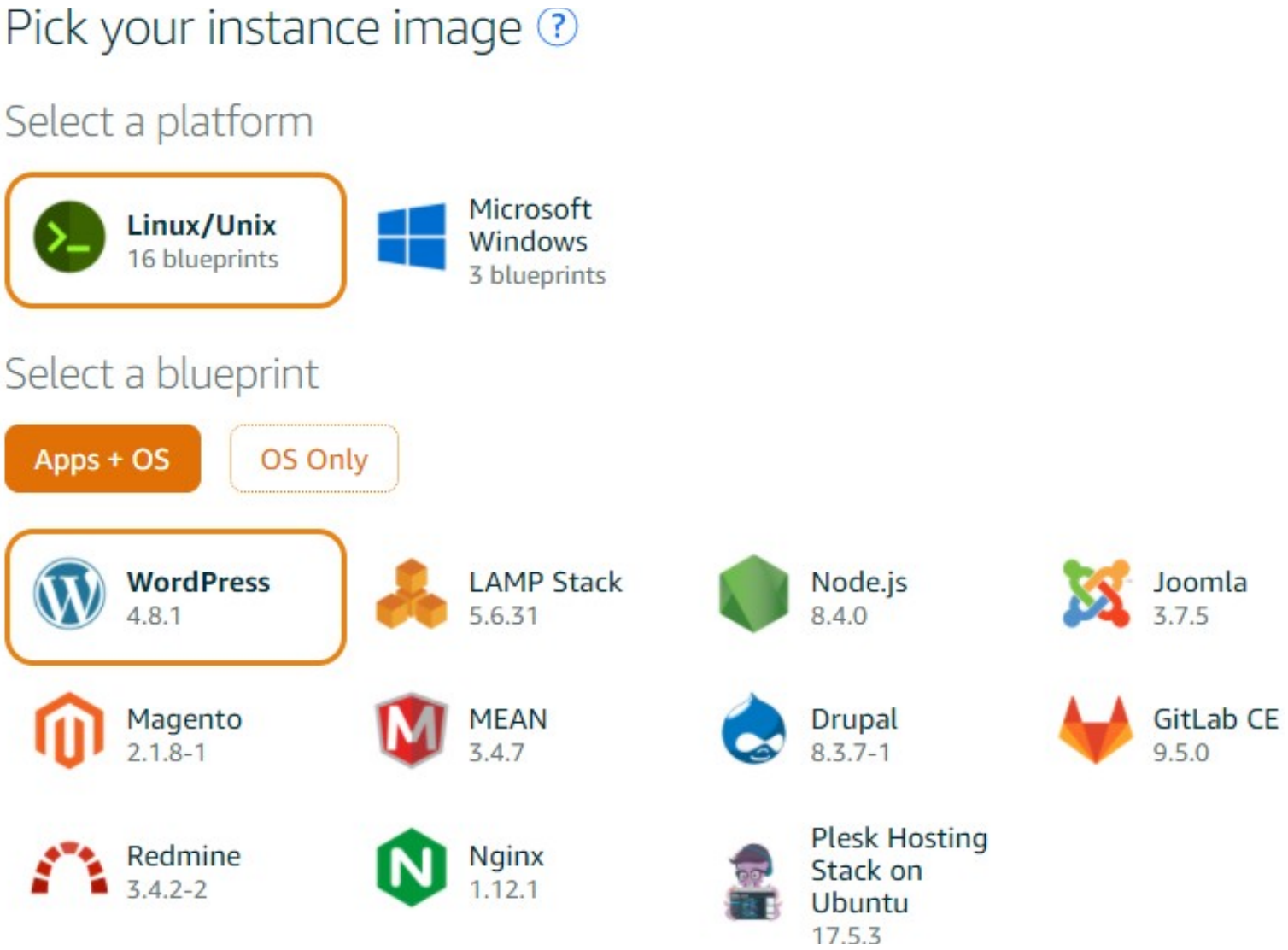
Amazon Lightsail

- Amazon Lightsail is the easiest way to get started with AWS if you just need virtual private servers
- It includes everything you need to launch your project quickly for a low, predictable price
  - Virtual machines
  - SSD-based storage
  - Data transfer
  - DNS management
  - Static IPs

Lightsail Features

- You can manage your instances using the Lightsail console, Lightsail API, or Lightsail command line interface (CLI)
- Lightsail is for developers
  - You can choose an image for your Lightsail instance that jumpstarts your dev project so you don't have to spend as much time installing software or frameworks
- Lightsail has images with base operating systems, development stacks like LAMP, LEMP (Nginx), and SQL Server Express, and applications like WordPress, Drupal, and Magento
- You can take snapshots of these instances and disks and easily create new instances from those snapshots
- You can also create a Lightsail load balancer and attach target instances to create a highly available application. You can also configure your load balancer to handle encrypted (HTTPS) traffic, session persistence, health checking, and more

Lightsail instances

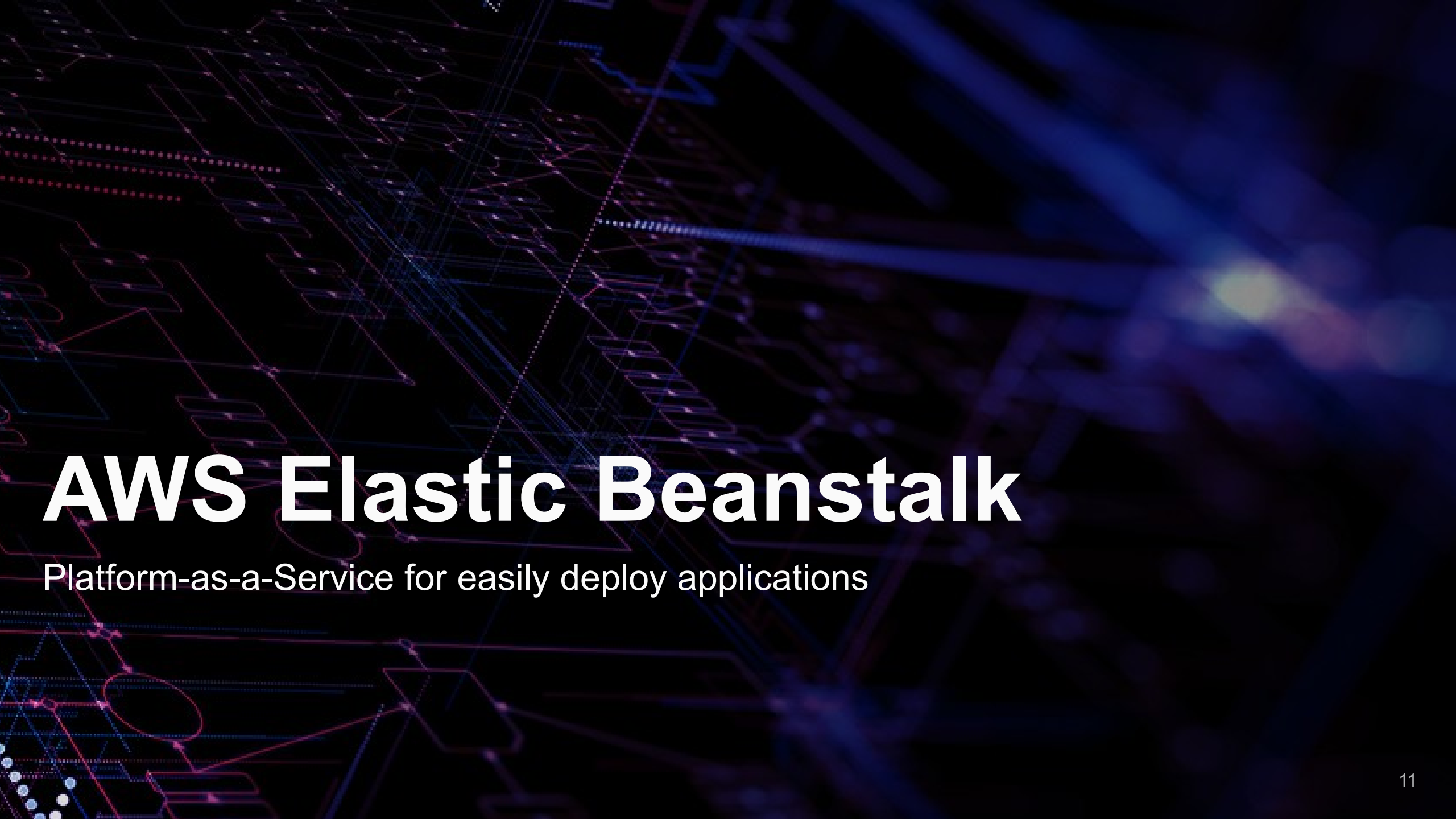


Lightsail Features

- Static IPs
  - The default dynamic public IP address attached to your Amazon Lightsail instance changes every time you stop and restart the instance
  - Create a static IP address and attach it to your instance to keep the public IP address from changing
- DNS Records
  - To map your domain name, such as example.com, to your Amazon Lightsail instance, you add a record to the Domain Name System (DNS) of your domain
  - You can manage domain DNS records at the registrar where the domain name was registered, or manage them separately using another DNS hosting provider, such as Lightsail
- Snapshots
  - After a Lightsail instance snapshot is exported and available in Amazon EC2 (as an AMI and an EBS snapshot), you can create an Amazon EC2 instance from the snapshot using the Create an Amazon EC2 instance page in the Amazon Lightsail console
  - Lightsail uses an AWS CloudFormation stack to create EC2 instances and their related resources

Lightsail Pricing

- Virtual Servers
  - Linux from $3.50/mo
  - Up to 8 cores, RAM 32 GB, 7 TB network transfer per $160/mo
  - Windows from $8/mo
  - Up to 8 cores, RAM 32 GB, 7 TB network transfer per $240/mo
- Managed Databases
  - From $15/mo ($30 for high availability)
  - Up to 2 cores, RAM 8 GB, SSD 240 GB per $115/mo
- Load Balancer per $18/mo
- Block storage with high availability, about 10 GB per dollar


AWS Elastic Beanstalk
Platform-as-a-Service for easily deploy applications

Elastic Beanstalk

- An easy-to-use service that deploys, manages and scales web applications and services
- It makes it easy to quickly deploy applications on AWS without having to worry about infrastructure
- It is a platform as a service (PaaS)
  - You have the whole infrastructure created for you, so you can just deploy your application
- Allows quick deployment of you applications
  - Any code previously written, on some specific language, it can be simply placed it over the platform
- Reduces management complexity
  - You do not need to worry about managing the whole system, but if you wish you can have full control

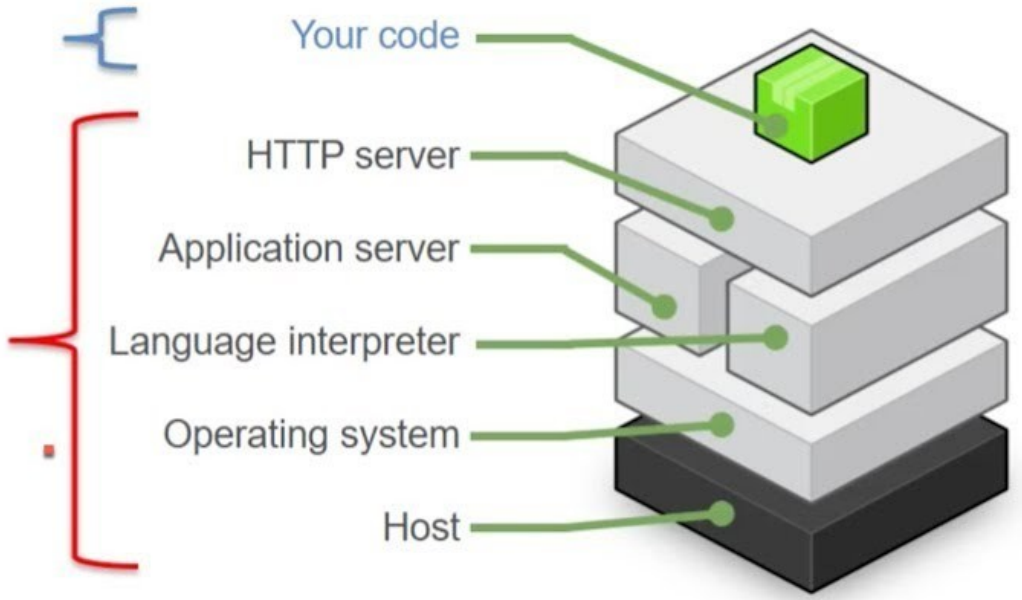
PaaS Schema

Elastic Beanstalk

On-instance configuration

Focus on building your application

Elastic Beanstalk configures each Amazon EC2 instance in your environment with the components necessary to run applications for the selected platform. No more worrying about logging into instances to install and configure your application stack.



Provided by you

Provided and managed by Elastic Beanstalk

Elastic Beanstalk: keep control in your hands

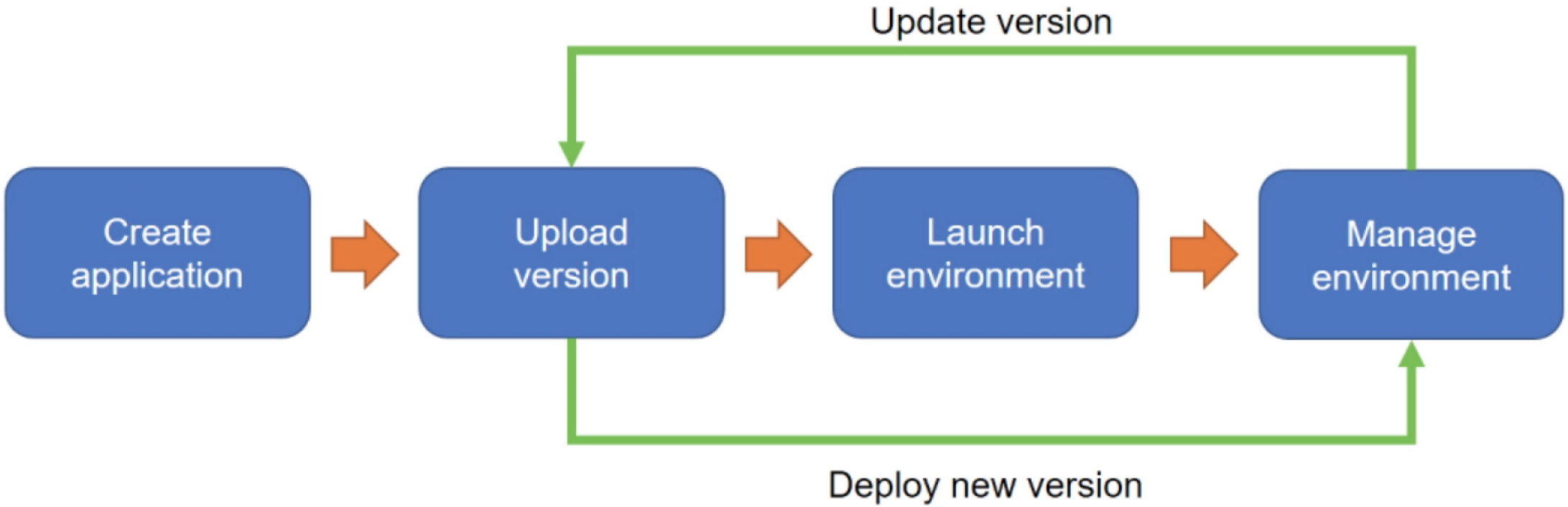
- Steps to have your application running on the cloud:

1. Create the application
2. Deploy the application
3. Elastic Beanstalk provisions capacity, set up load balancing and auto scaling, and configures monitoring
   - You can choose your instance type and your database
   - Set and adjust auto scaling
   - Access server log files
   - Enable HTTPS on load balancer
4. You can then manage your environment and deploy new application versions
   - Update your application easily

Elastic Beanstalk platforms

- Languages
  - Java, .NET, PHP, Node.js, Python, Ruby, Go
- Webservers and Containers
  - Apache, IIS, Java SE, Nginx, Passenger, Puma, Tomcat, Docker
- Deployment Platforms
  - Git, IDEs (Eclipse, Visual Studio), manual upload (WAR files, ZIP)
- AWS integration
  - EC2, S3, SNS, ELB, Auto Scaling

Elastic Beanstalk deployment and updates



- Environment: version that is deployed with AWS resources
- Environment configuration: settings and parameters that define the environment and resources
- Configuration template: used to create repeatable environment configurations

Elastic Beanstalk Pricing

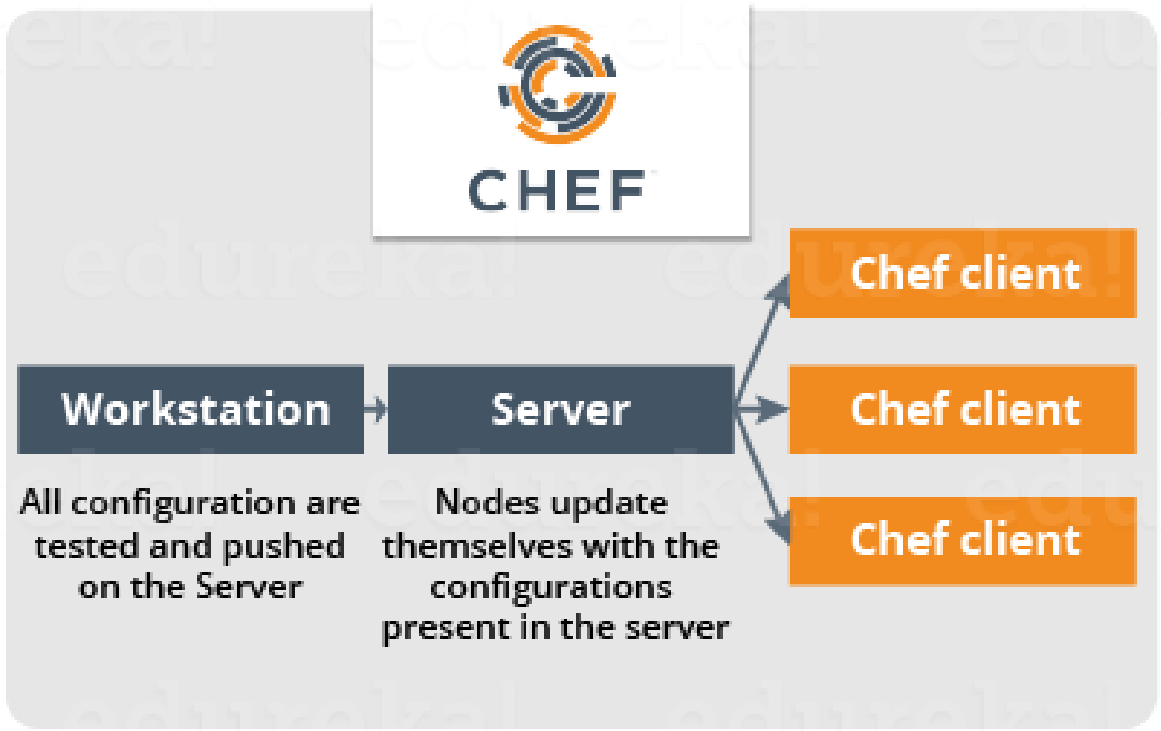
- There is no additional charge for AWS Elastic Beanstalk
- You pay for the AWS resources you create to store and run your application
  - EC2 instances
  - S3 buckets
  - Database storage
  - ELB
  - …
- You only pay for what you use, as you use it
  - There are no minimum fees and no upfront commitments

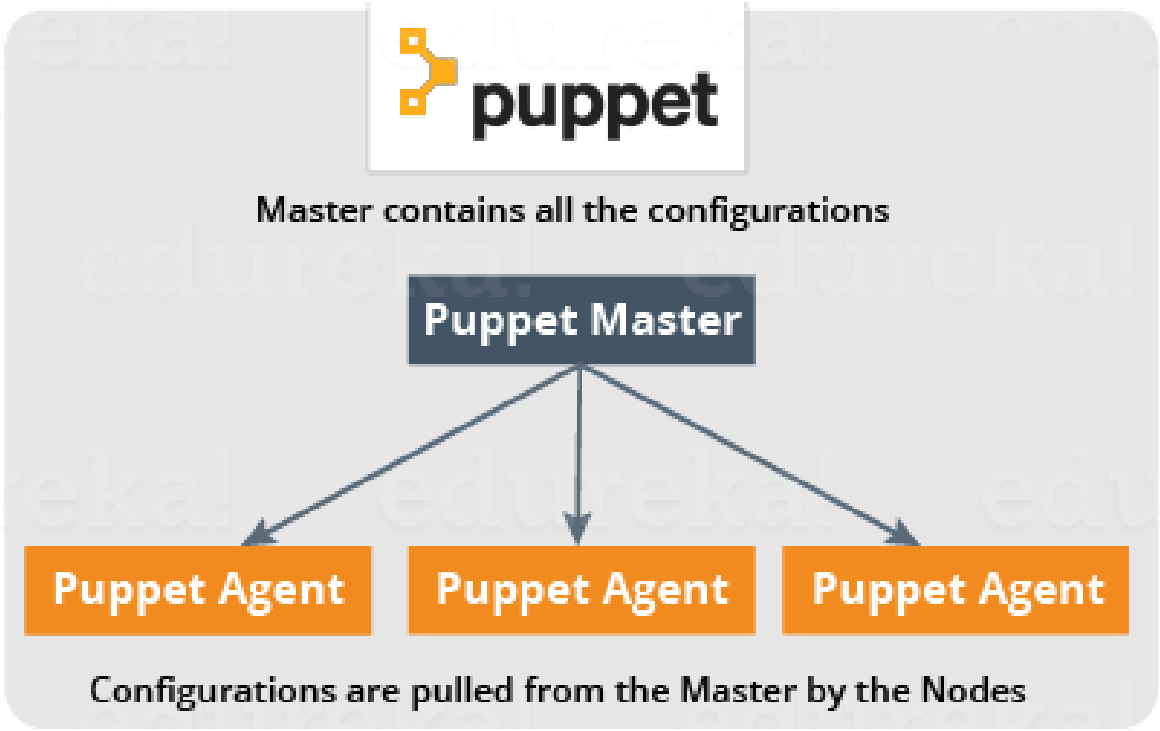

AWS OpsWorks
Configuration Management with Chef and Puppet

OpsWorks

- Chef and Puppet are automation platforms that allow you to use code to automate the configurations of your servers
- AWS OpsWorks is a configuration management service that provides managed instances of Chef and Puppet
- OpsWorks lets you use Chef and Puppet to automate how servers are configured, deployed, and managed across your Amazon EC2 instances or on-premises compute environments
- OpsWorks has three offerings
  - AWS Opsworks for Chef Automate
  - AWS OpsWorks for Puppet Enterprise
  - AWS OpsWorks Stacks

Chef and Puppet





OpsWorks for Chef Automate

- AWS OpsWorks for Chef Automate provides a fully managed Chef Automate server and suite of automation tools
- That gives you workflow automation for continuous deployment, automated testing for compliance and security, and a user interface that gives you visibility into your nodes and their status
- The Chef Automate platform handles operational tasks such as software and operating system configurations, continuous compliance, package installations, database setups, and more
- The Chef server centrally stores your configuration tasks and provides them to each node in your compute environment at any scale, from a few nodes to thousands of nodes
- It is completely compatible with tooling and cookbooks from the Chef community and automatically registers new nodes with your Chef server

OpsWorks for Puppet Enterprise

- AWS OpsWorks for Puppet Enterprise is a fully managed configuration management service that hosts Puppet Enterprise, a set of automation tools from Puppet for infrastructure and application management
- OpsWorks also maintains your Puppet master server by automatically patching, updating, and backing up your server
- It eliminates the need to operate your own configuration management systems or worry about maintaining its infrastructure
- OpsWorks gives you access to all of the Puppet Enterprise features, which you manage through the Puppet console
- It also works seamlessly with your existing Puppet code

OpsWorks Stacks

- AWS OpsWorks Stacks lets you manage applications and servers on AWS and on-premises
- You can model your application as a stack containing different layers, such as load balancing, database, and application server
- You can deploy and configure Amazon EC2 instances in each layer or connect other resources such as Amazon RDS databases
- OpsWorks Stacks lets you set automatic scaling for your servers based on preset schedules or in response to changing traffic levels, and it uses lifecycle hooks to orchestrate changes as your environment scales
- You run Chef recipes using Chef Solo, allowing you to automate tasks such as installing packages and programming languages or frameworks, configuring software, and more

AWS CloudFormation

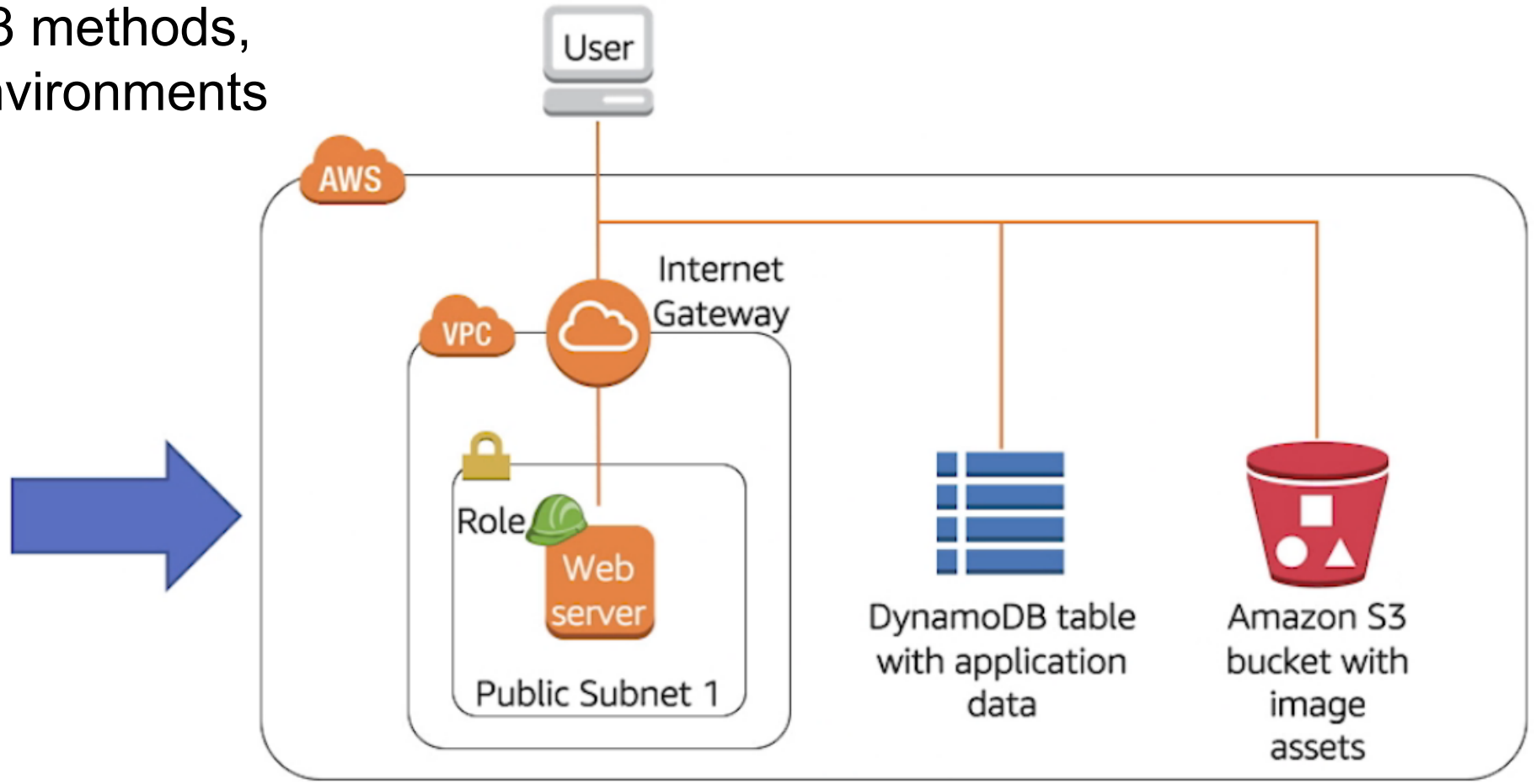
Infrastructure as code at a glance

Provisioning AWS resources

- How do we interact with services? We call APIs and tell the services what to do

➢Using any of these 3 methods, we can construct environments for our workloads

AWS…
- Management console
- CLI
- SDK / API

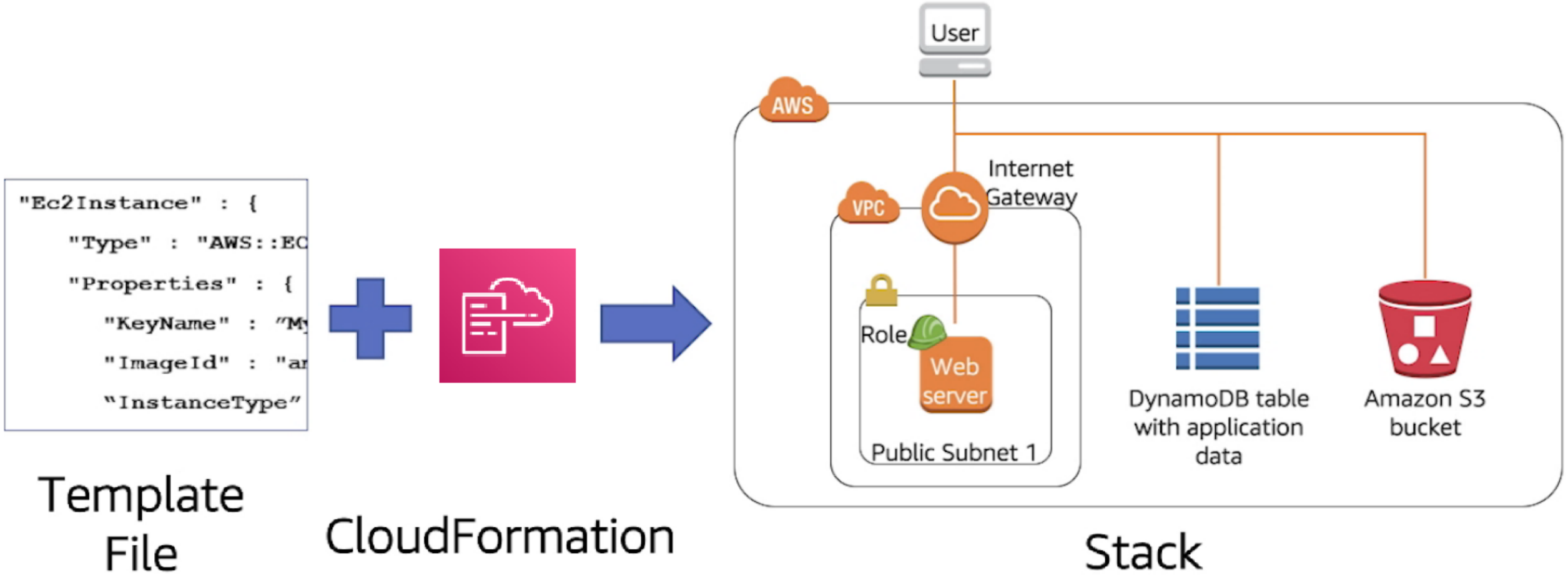


CloudFormation

- Automates resource provisioning by simplifying the task of repeatedly and predictably creating groups of related resources that power your applications
- Provides a common language for you to describe and provision all the infrastructure resources in your cloud environment
- Provides and manages stacks of AWS resources based on templates you create to model your infrastructure architecture
- Allows you to use a simple text file to model and provision, in an automated and secure manner, all the resources needed for your applications across all regions and accounts
  - This file serves as the single source of truth for your cloud environment
- Available at no additional charge, and you pay only for the AWS resources needed to run your applications

CloudFormation management

- You create a template file that CloudFormation uses to create your stack



CloudFormation template

- A JSON or YAML formatted text file that CloudFormation uses as blueprints for building your AWS resources
- You can save these files with any extension, such as .json, .yaml, .template, or .txt
- Example: CloudFormation provisions an instance with:
  - ✓ An ami-0ff8a91507f77f867 AMI ID
  - ✓ A t2.micro instance type
  - ✓ A testkey key pair name
  - ✓ An EBS volume

  And an elastic IP@

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "A sample template",
  "Resources" : {
    "MyEC2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : "ami-0ff8a91507f77f867",
        "InstanceType" : "t2.micro",
        "KeyName" : "testkey",
        "BlockDeviceMappings" : [
          {
            "DeviceName" : "/dev/sdm",
            "Ebs" : {
              "VolumeType" : "io1",
              "Iops" : "200",
              "DeleteOnTermination" : "false",
              "VolumeSize" : "20"
            }
          }
        ]
      }
    },
    "MyEIP" : {
      "Type" : "AWS::EC2::EIP",
      "Properties" : {
        "InstanceId" : {"Ref": "MyEC2Instance"}
      }
    }
  }
}
```

CloudFormation stacks

- Resources generated by the template you created
- Units of deployment where you can create, update, and delete a collection of resources by creating, updating, and deleting stacks
  - When you delete a stack, all resources in the stack are deleted
- With templates and stacks you have an example of Infrastructure as a Code, where you use code to establish your infrastructure
- Example: Suppose you created a template that includes an Auto Scaling group, ELB load balancer, and an RDS database instance. To create those resources, you create a stack by submitting the template that you created, and CloudFormation provisions all those resources for you

Elastic Beanstalk vs CloudFormation

- **Elastic Beanstalk**
- Provides an environment to easily deploy and run applications in the cloud
- Integrated with developer tools and provides a one-stop experience for you to manage the lifecycle of your applications
- PaaS

- **CloudFormation**
- A provisioning mechanism for a broad range of AWS resources
- Supports the infrastructure needs of many different types of applications
- Supports Elastic Beanstalk application environments as one of the AWS resource types
- IaaS & Infrastructure as Code

Provisioning & Deployment documentation

AWS Elastic Beanstalk
- https://aws.amazon.com/elasticbeanstalk/

AWS CloudFormation
- https://docs.aws.amazon.com/cloudformation/

Amazon Lightsail
- https://lightsail.aws.amazon.com/ls/docs

AWS OpsWorks
- https://aws.amazon.com/opsworks/