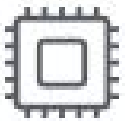
AWS SERVICES OVERVIEW

AWS Compute Services



Compute

EC2

Elastic Beanstalk

ECR

ECS

EKS

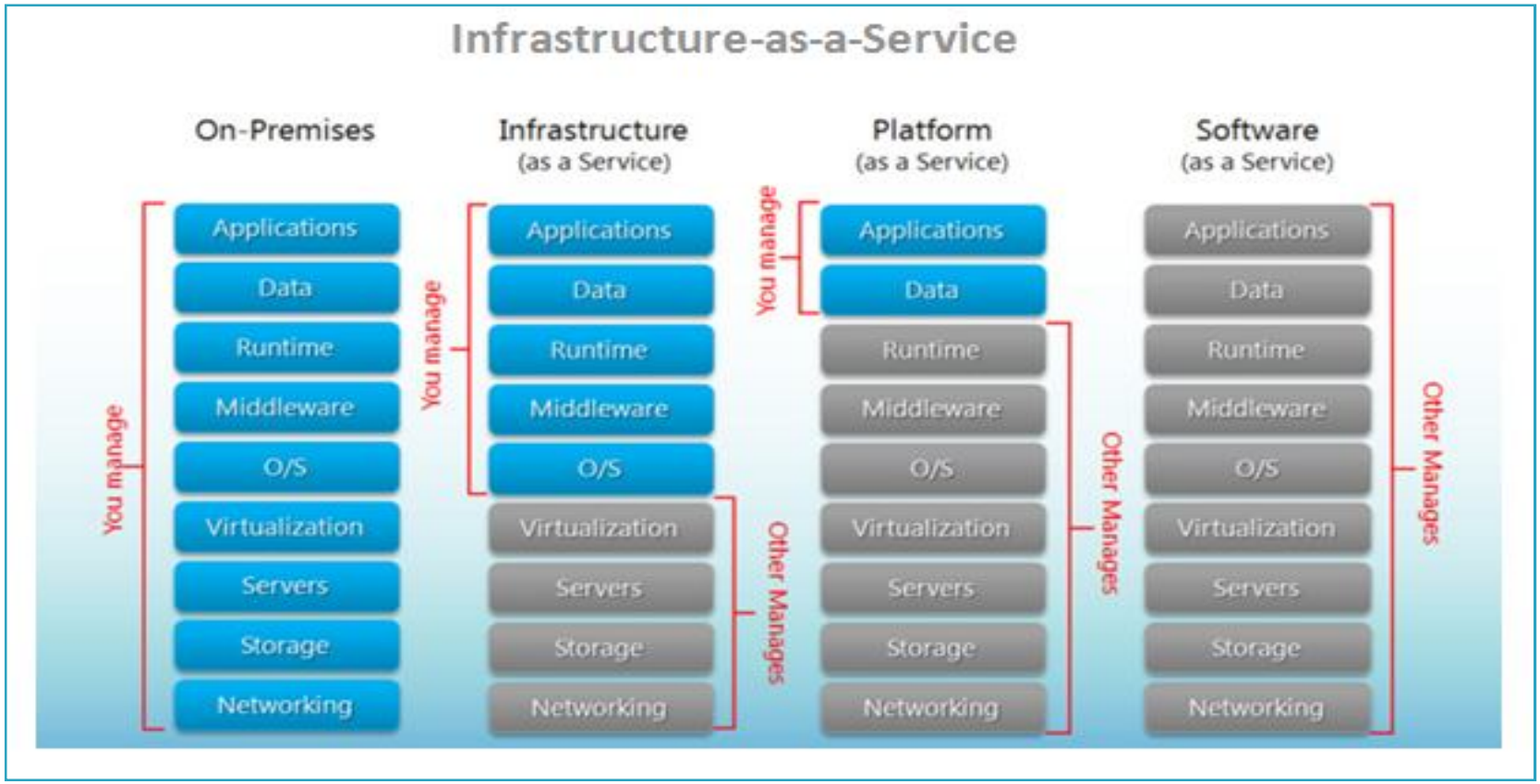
Lambda

Batch

Lightsail

EC2 (Elastic Compute Cloud)

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud



EC2 Overview

Amazon EC2

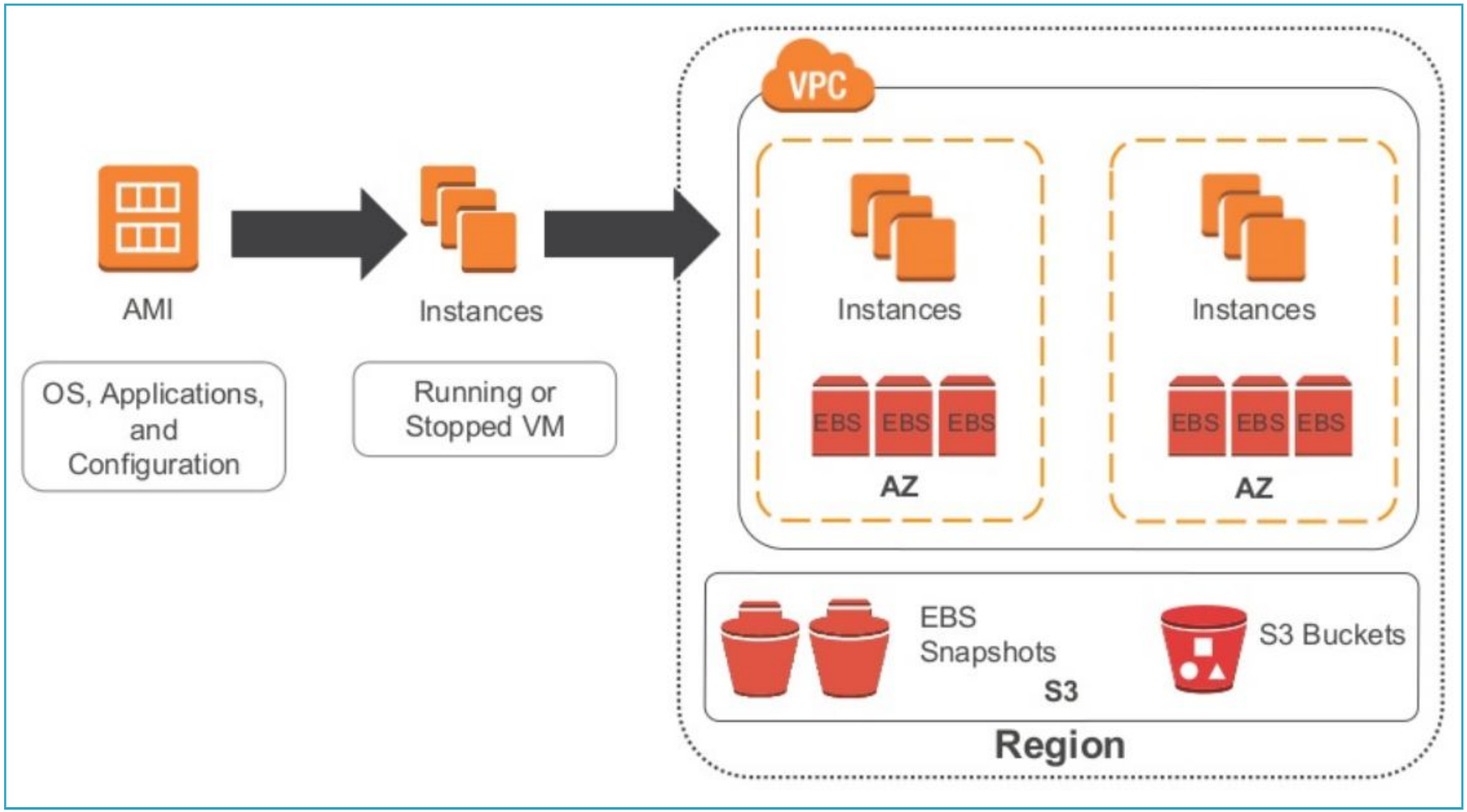
- **Resizable** compute capacity
- Complete control of your computing resources
- **Reduced time required** to obtain and boot new server instances

- **Scale capacity** as your computing requirements change
- Pay only for capacity that you actually use
- Choose **Linux** or **Windows**
- Deploy across **AWS Regions** and **Availability Zones** for reliability
- Use **tags** to help manage your Amazon EC2 resources

EC2 Concepts

- **Instance -** Virtual Computing environments
- **AMI -** Preconfigured template of instance
- **Instance Types** - Various configs of CPU, memory, storage and network
- **Key Pair** – Secure login to instance
- **Instance Store Volumes** – Temporary Data Storage
- **EBS (Elastic Block Store)** – Persistent Storage Volume
- **Availability Zones** – Physical locations of resources
- **Security Groups** – Firewall to protect resource access based on rules
- **Placement Groups -** Determines how instances are placed on underlying hardware
- **Tags** – Metadata about the instance

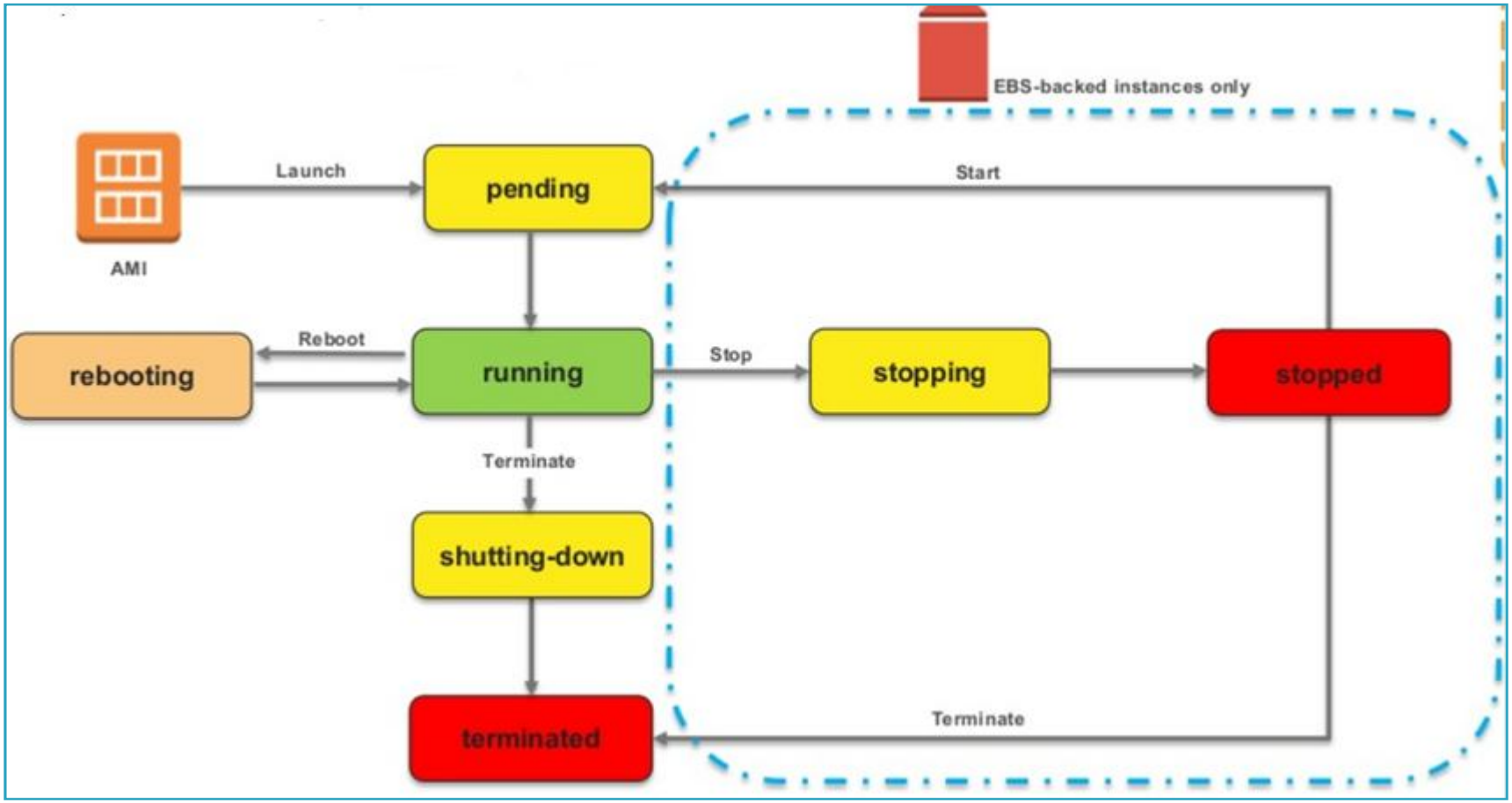
EC2 Instances



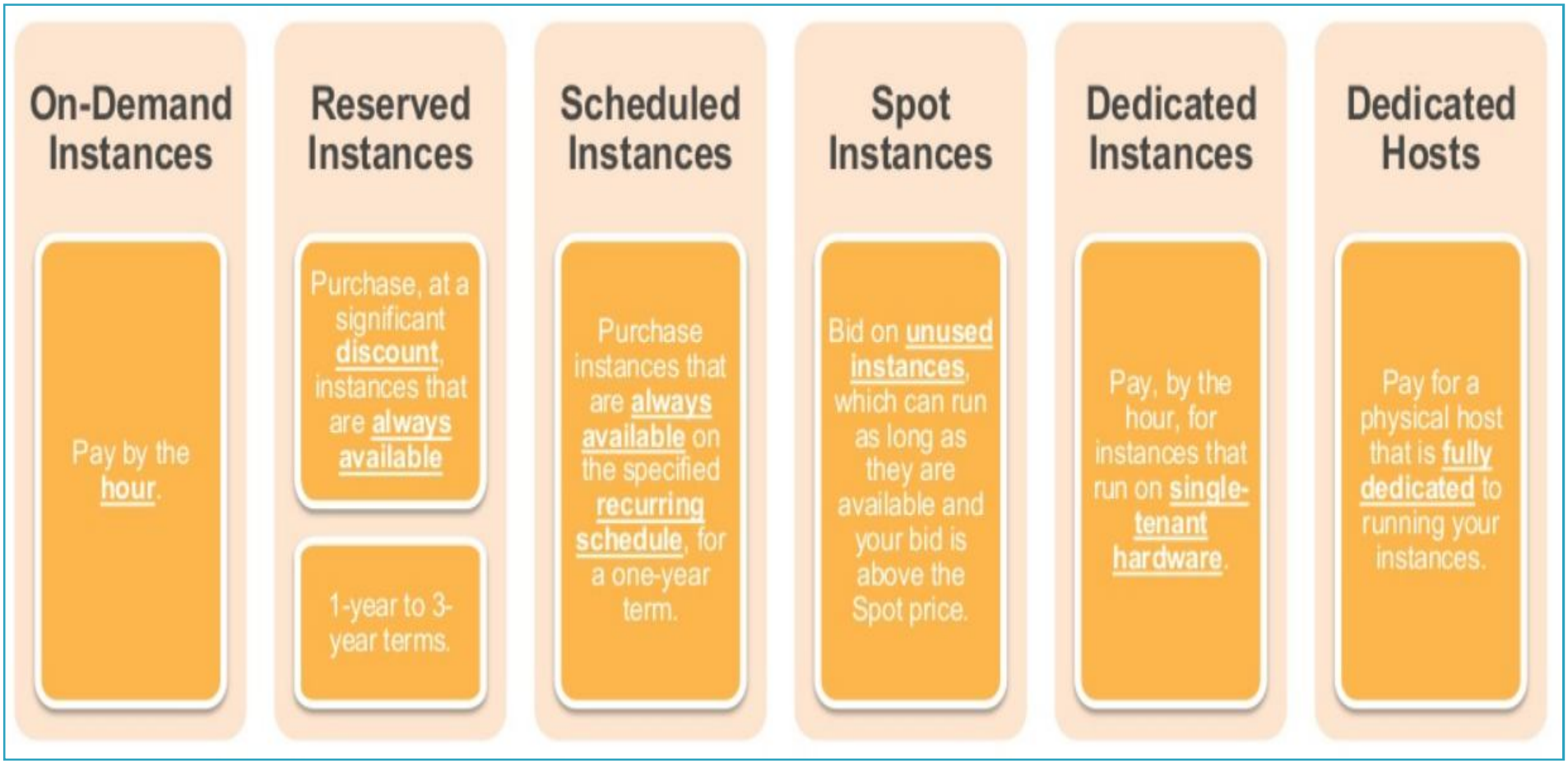
EC2 Instance Types

Instance Family	Some Use Cases
General purpose (t2, m4, m3)	• Low-traffic websites and web applications • Small databases and mid-size databases
Compute-optimized (c4, c3)	• High performance front-end fleets • Video-encoding
Memory-optimized (r3)	• High performance databases • Distributed memory caches
Storage-optimized (i2, d2)	• Data warehousing • Log or data-processing applications
GPU instances (g2)	• 3D application streaming • Machine learning

Instance Lifecycle

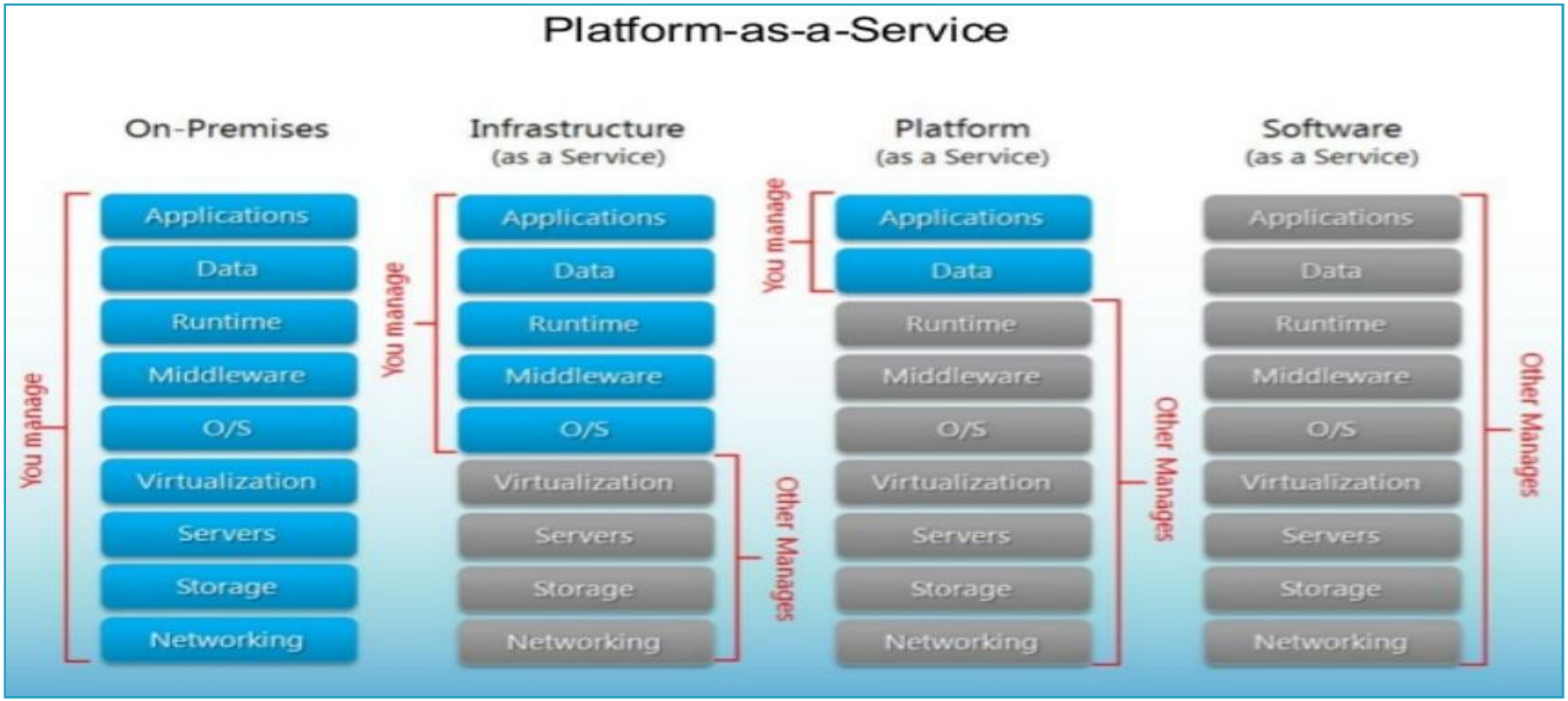


Instance Purchasing Options



Elastic Beanstalk

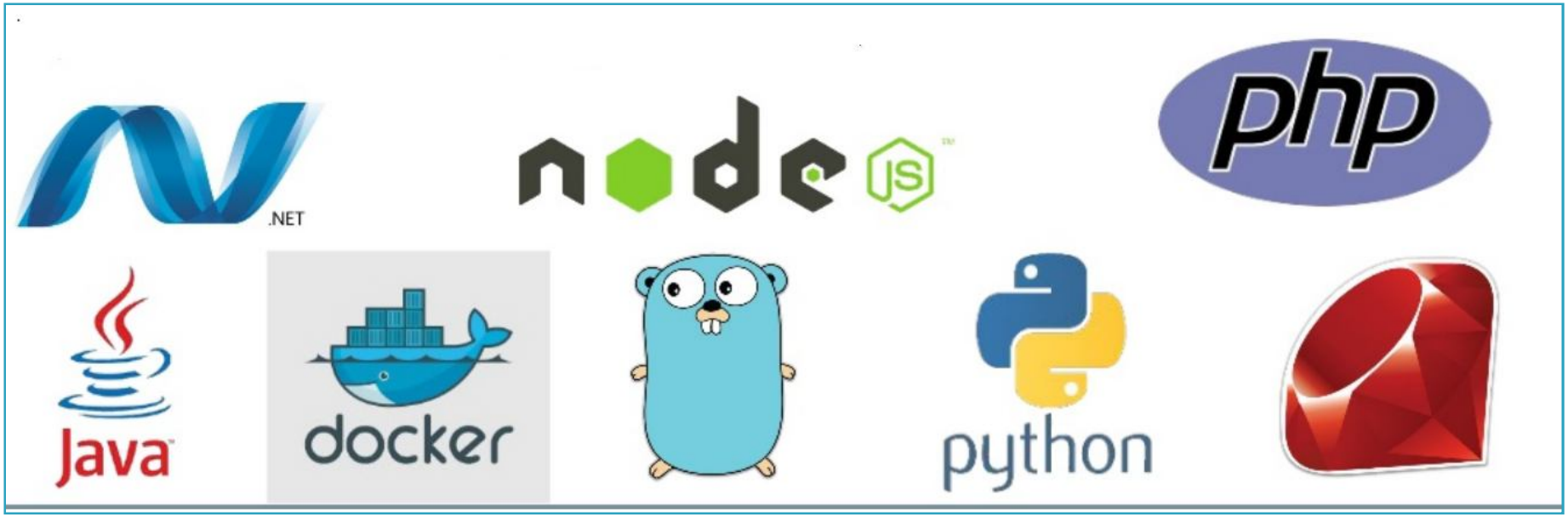
ElasticBeanstalk enables to quickly deploy and manage applications in the AWS Cloud without worrying about the infrastructure that runs those applications. Reduces management complexity without restricting choice or control.



Elastic Beanstalk concepts

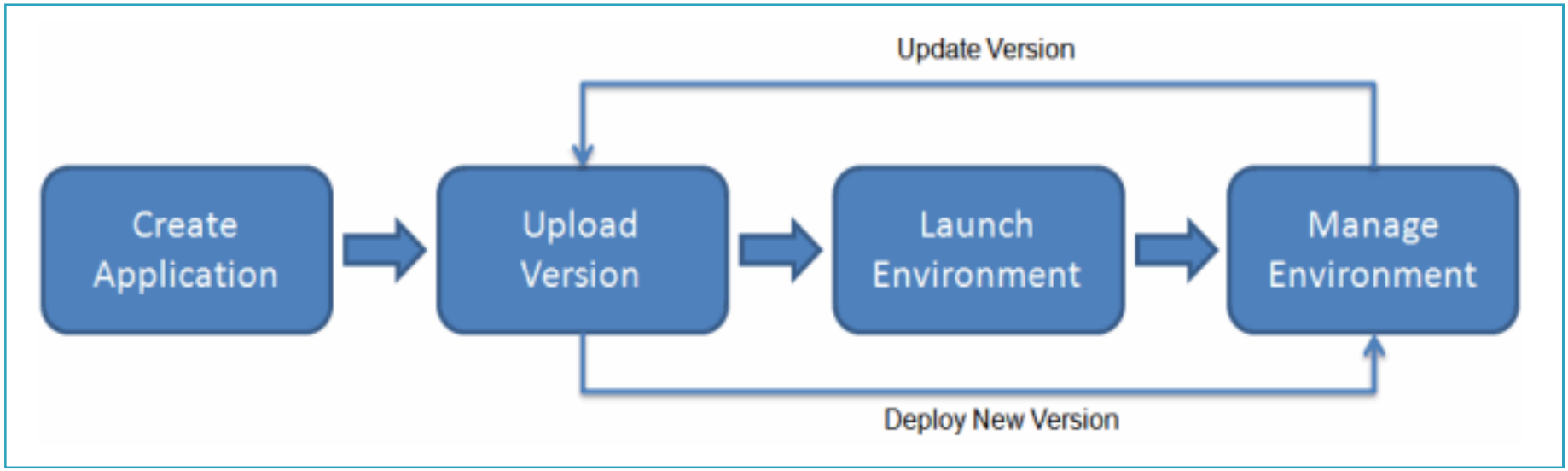
- **Application -** Logical collection of Beanstalk components (env, version, config)
- **Application Version -**Specific, labeled iteration of deployable code
- **Environment** - Provisions the resources needed to run the application version
- **Environment Tier** – Web server or Worker environment
- **Environment Config** – Collection of parameters or settings to define behavior
- **Configuration Template** – Blueprint of environment configuration

Elastic Beanstalk Launguage Support

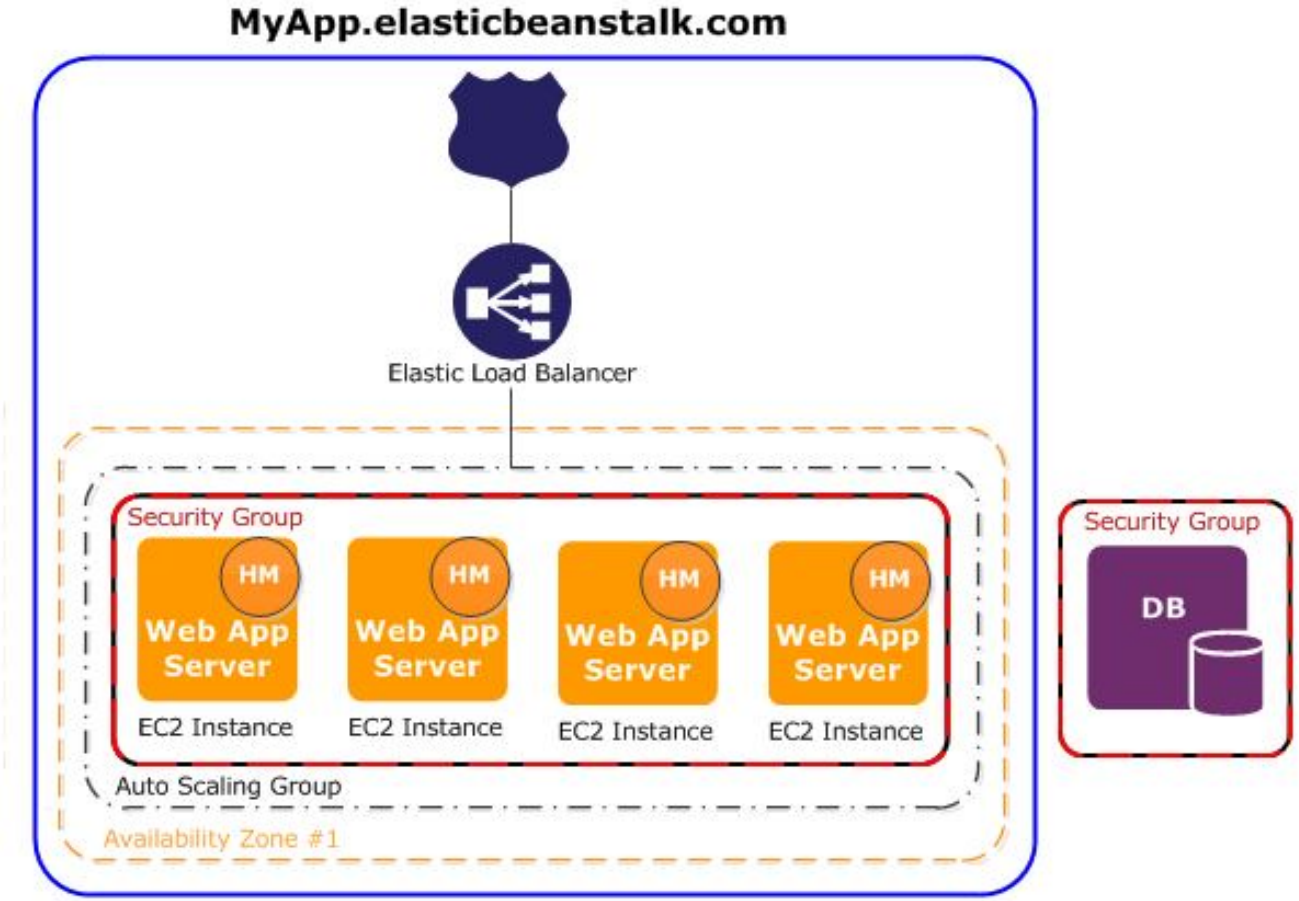


Deployment Artifacts: zip, war, docker

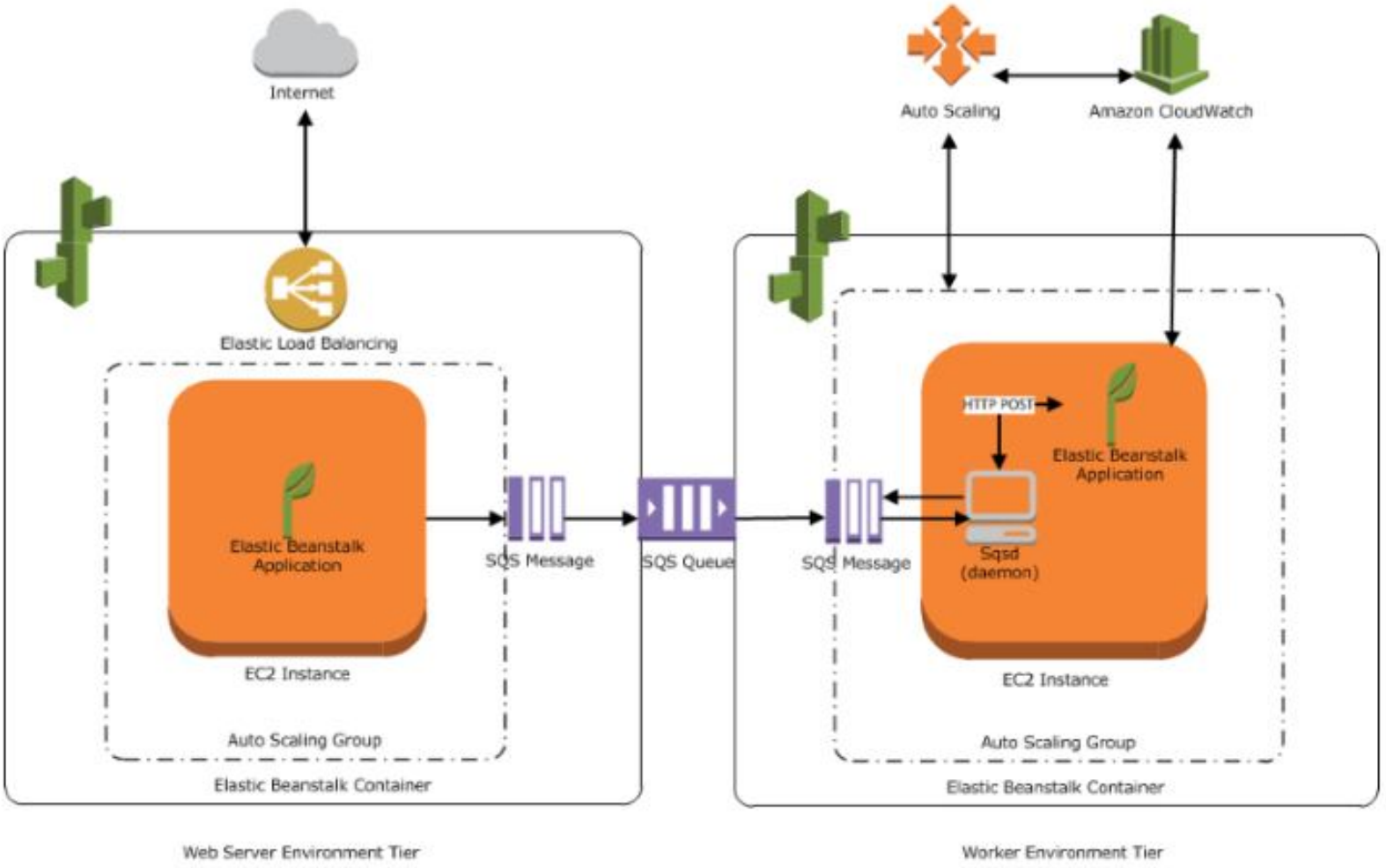
Elastic Beanstalk Workflow



Elastic Beanstalk Architecture (Web)



Elastic Beanstalk Architecture (Worker)



Elastic Beanstalk Competitors

- Cloud Foundry
- Google App Engine
- Azure Web Sites
- Heroku
- Bluemix

Elastic Load Balancer

Elastic Load Balancer

Elastic Load Balancing

- **Distributes** traffic across multiple EC2 instances, in multiple Availability Zones
- Supports **health checks** to detect unhealthy Amazon EC2 instances
- Supports the **routing and load balancing** of HTTP, HTTPS, SSL, and TCP traffic to Amazon EC2 instances

Elastic Load Balancer Types

Classic Load Balancer

PREVIOUS GENERATION
for HTTP, HTTPS, and TCP

Choose a Classic Load Balancer when you have an existing application running in the EC2-Classic network.

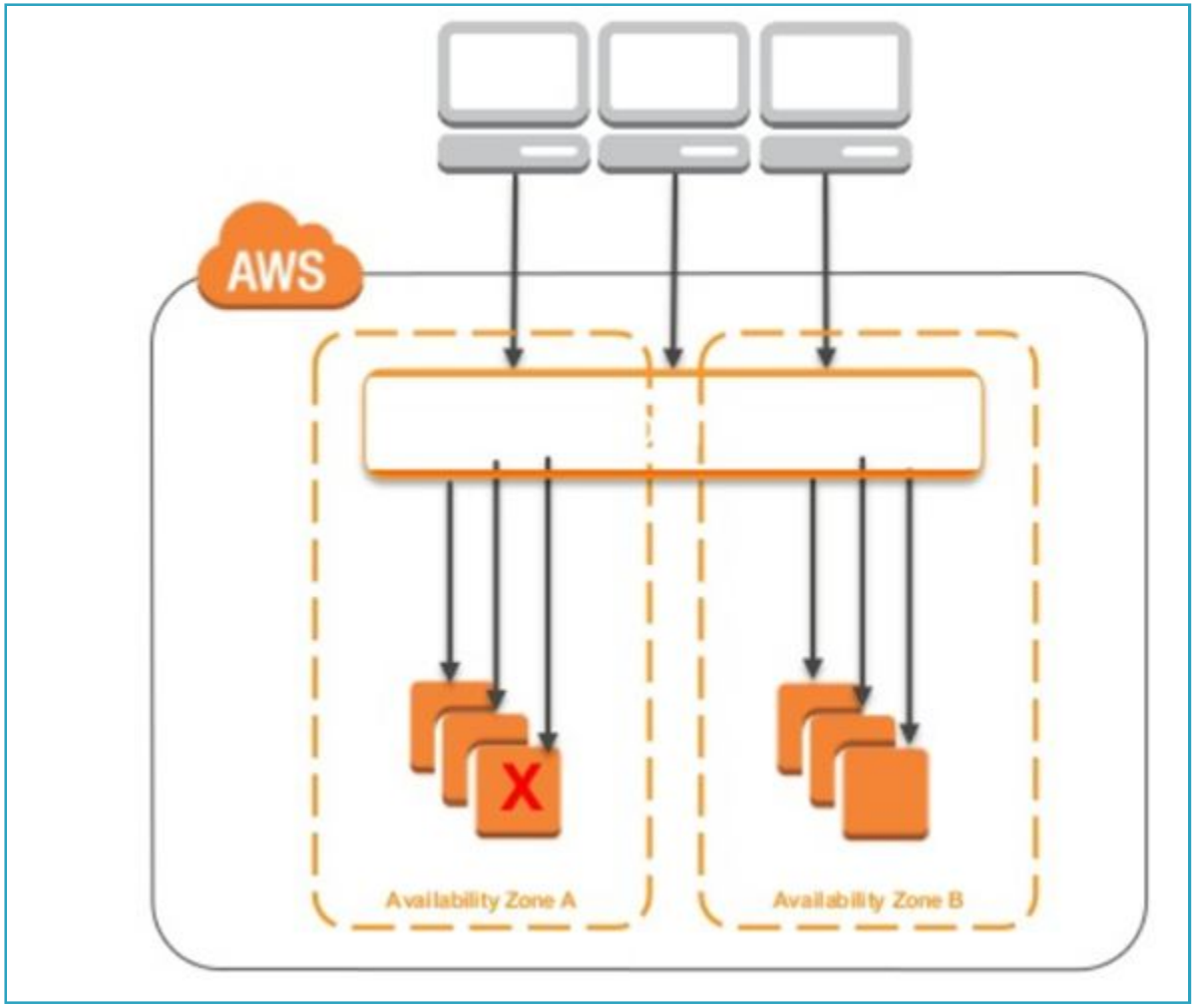
Application Load Balancer

Choose an Application Load Balancer when you need a flexible feature set for your web applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Network Load Balancer

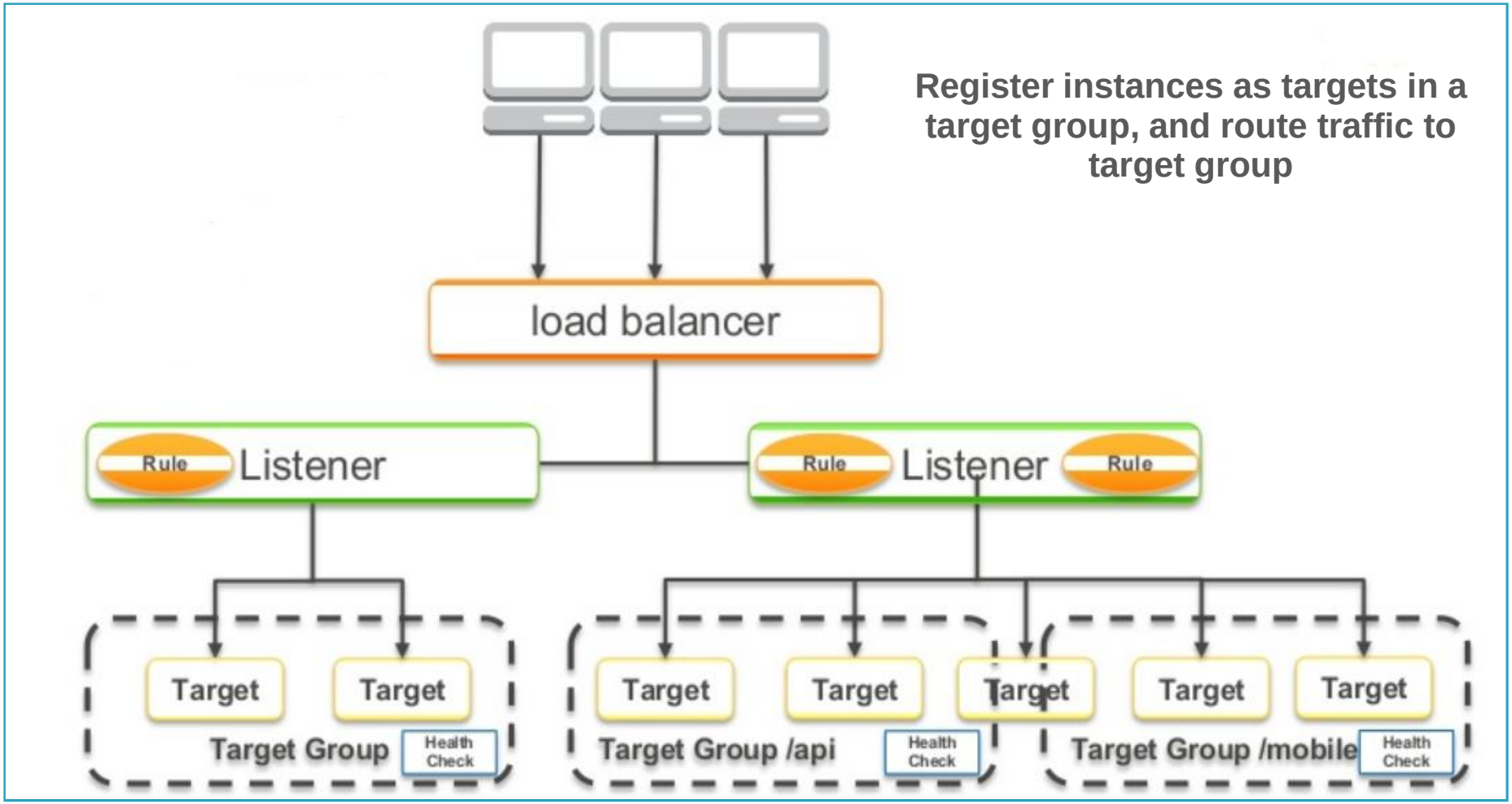
Choose a Network Load Balancer when you need ultra-high performance, the ability to terminate TLS connections at scale, centralize certificate deployment, and static IP addresses for your application. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Classic Load Balancer



Register instance with your load balancer

Application Load Balancer



Classic vs Application Load Balancer

Classic Load Balancer benefits include support for:

- EC2-Classic.
- VPC.
- TCP and SSL listeners.
- Sticky sessions.
- OSI Layer 4
  (network protocol level)

ALB benefits include support for:

- Path-based routing.
- Routing requests to multiple services on a single EC2 instance.
- Containerized applications.
- Monitoring the health of each service independently.
- OSI Layer 7
  (application level)

Auto Scaling

Auto Scaling

Auto Scaling

- **Scale** your Amazon EC2 capacity **automatically**
- Well-suited for applications that experience **variability in usage**
- Available at no additional charge

Auto Scaling Benefits

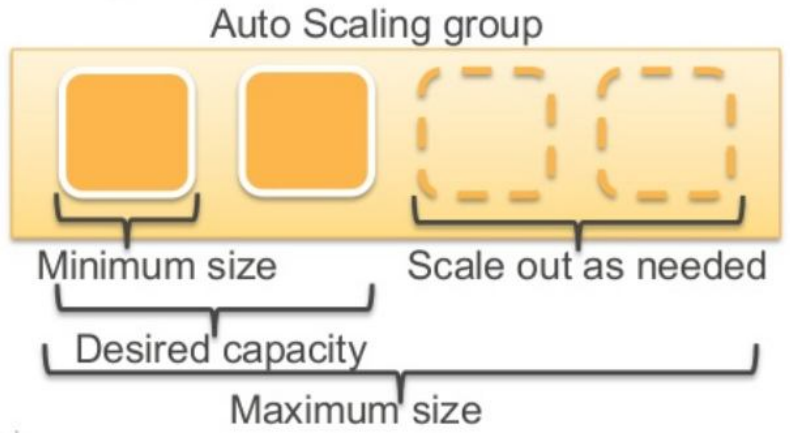
Better Fault Tolerance

Better Availability

Better Cost Management

Auto Scaling Groups

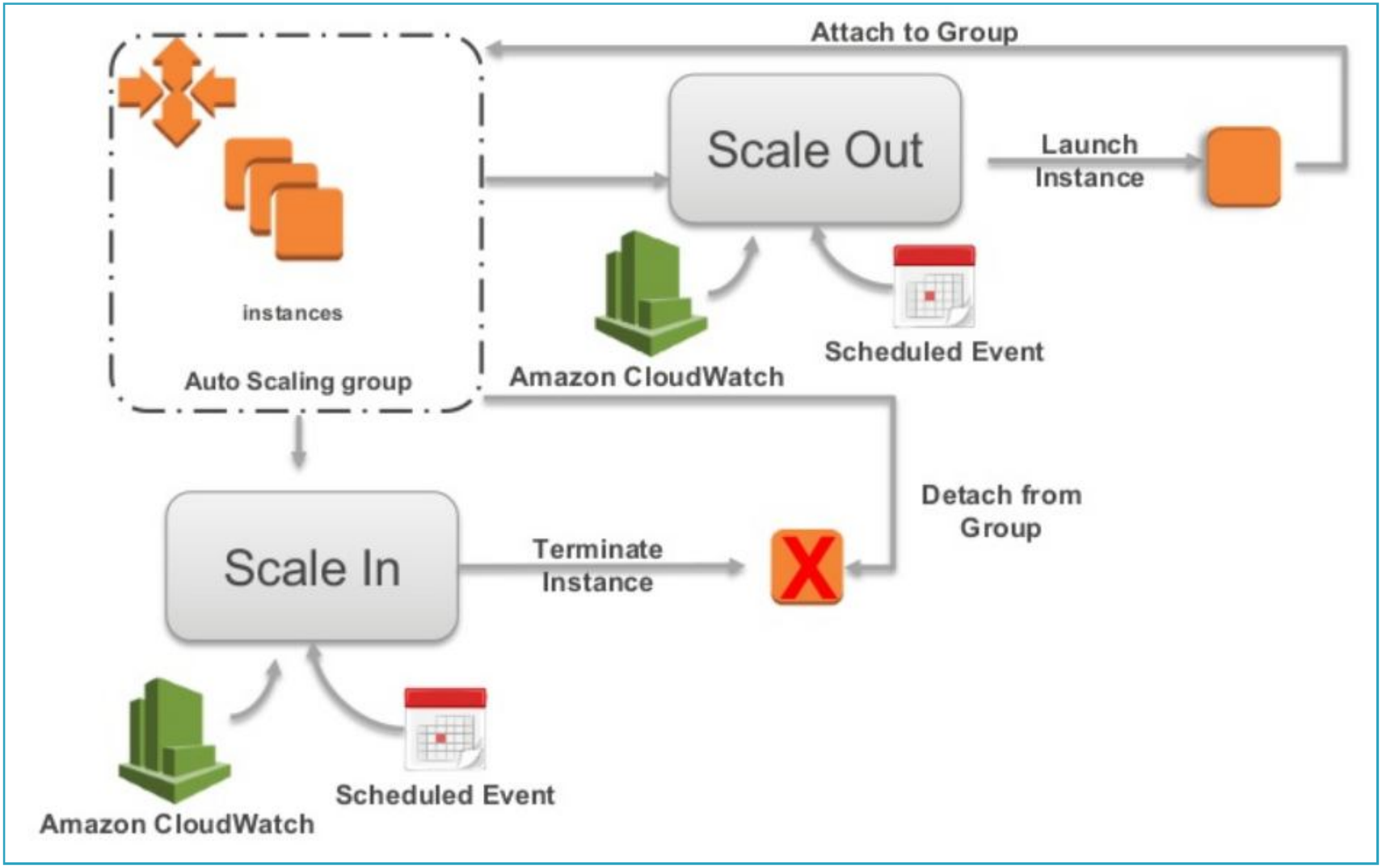
- Contain a collection of EC2 instances that share similar characteristics.
- Instances in an Auto Scaling group are treated as a **logical grouping** for the purpose of instance scaling and management.



Dynamic Scaling

- You can create a scaling policy that uses **CloudWatch alarms** to determine:
  - When your Auto Scaling group should **scale out**.
  - When your Auto Scaling group should **scale in**.
- You can use alarms to monitor:
  - Any of the metrics that AWS services send to Amazon CloudWatch.
  - Your own **custom metrics**.

Auto Scaling Lifecycle



AWS Storage Services

Storage

Amazon Simple Storage Service (S3)

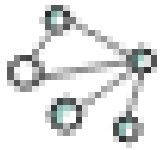
Amazon Elastic Block Storage (EBS)

Amazon Elastic File System (EFS)

Amazon Glacier

AWS Storage Gateway

AWS Network Services



Networking & Content Delivery

VPC

CloudFront

Direct Connect

Route 53

AWS Database Services

Database

RDS

DynamoDB

ElastiCache

Amazon Redshift

AWS Migration Services

Migration

AWS Migration Hub

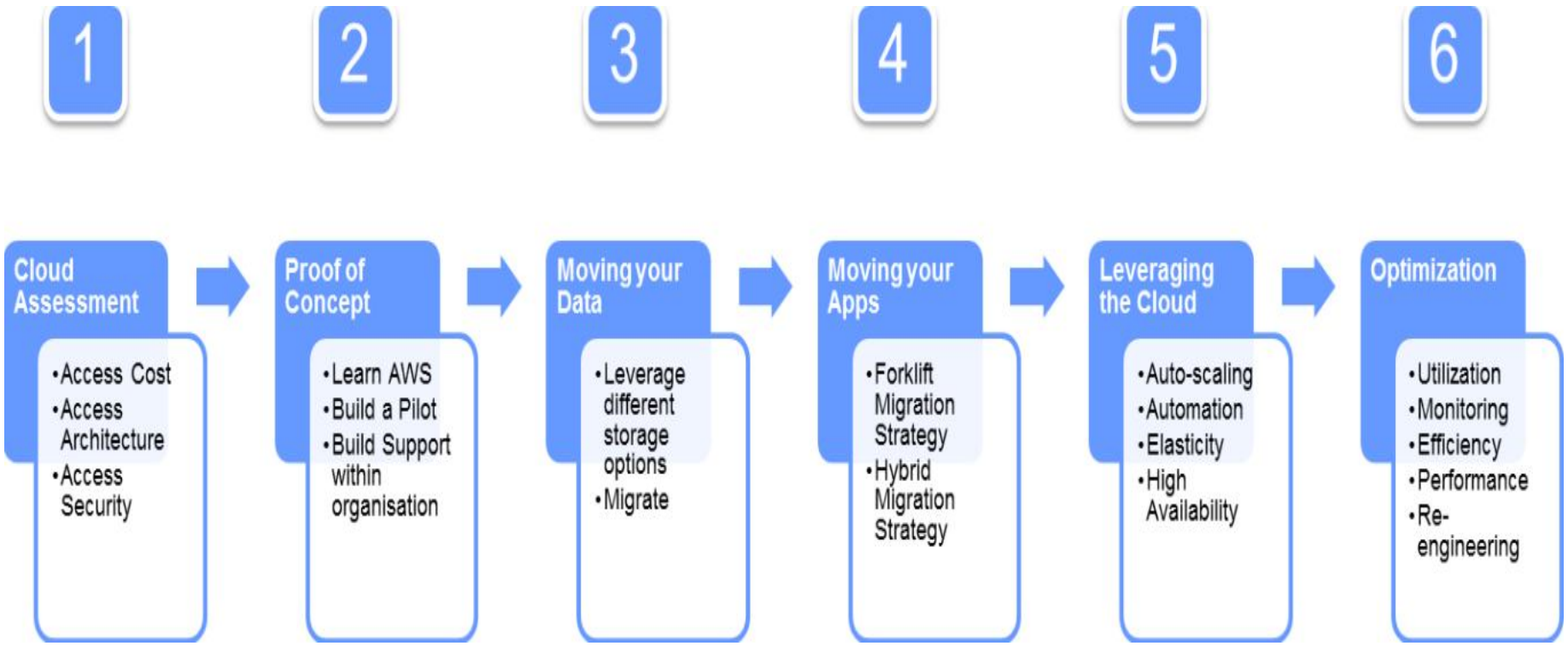
Application Discovery Service

Database Migration Service

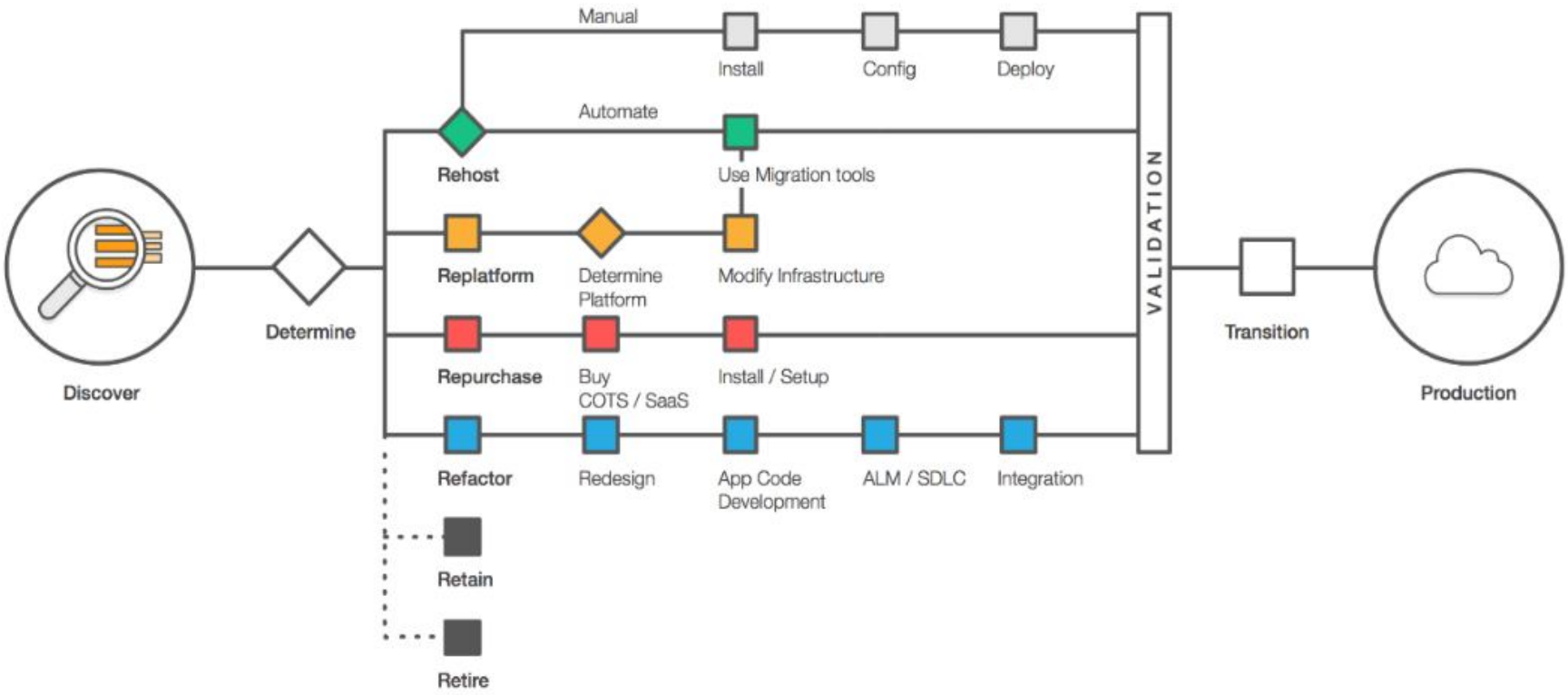
Server Migration Service

Snowball

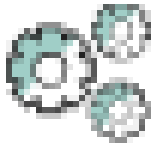
Cloud Migration Strategy



Cloud Migratrion Strategies



AWS Developer Services



Developer Tools

CodeStar

CodeCommit

CodeBuild

CodeDeploy

CodePipeline

X-Ray

AWS Messaging Services

Messaging

Simple Queue Service

Simple Notification Service

Simple Email Service

AWS Security Services

Security, Identity & Compliance

IAM

Inspector

Certificate Manager

Directory Service

WAF & Shield

Artifact

Amazon Macie

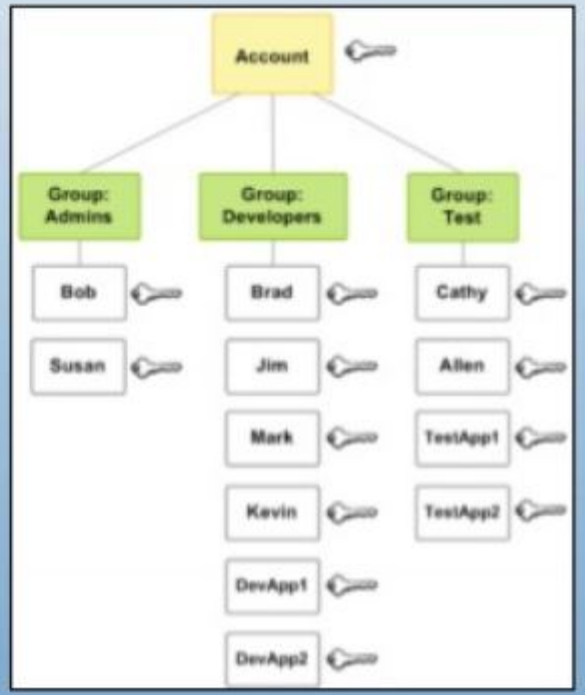
CloudHSM

AWS Identity and Access Management (IAM)

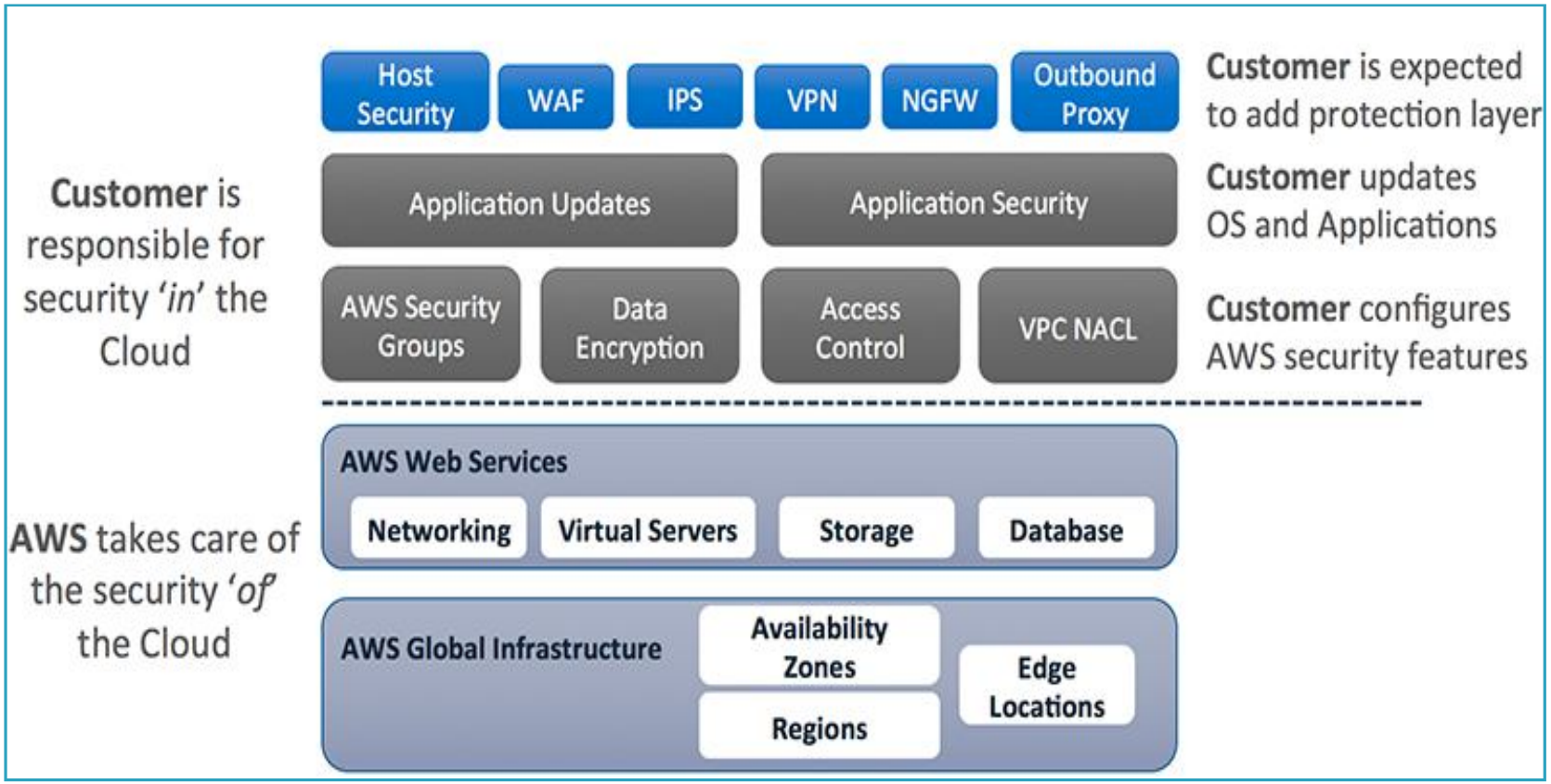
IAM is a web service for securely controlling access to AWS services. Centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users and applications can access.

AWS Identity and Access Management (IAM)

- Each account has root identity plus Users, Groups, Roles
  - Account-level: password complexity policies
- Unique security credentials for each user
  - Login/password (optional)
  - Access / secret keys (for APIs) (optional)
  - (V)MFA devices (optional)
- Policies control access to AWS APIs
- Deeper integration into some Services
  - S3: policies on objects and buckets
  - Simple DB: domains
- AWS Management Console supports IAM user log on
- Not for Operating Systems or Applications
  - use LDAP, Active Directory/ADFS, etc…



AWS Shared Security Model



AWS Security Compliance

AWS Management Services

Management Tools

CloudWatch

CloudFormation

CloudTrail

Config

OpsWorks

Service Catalog

Trusted Advisor

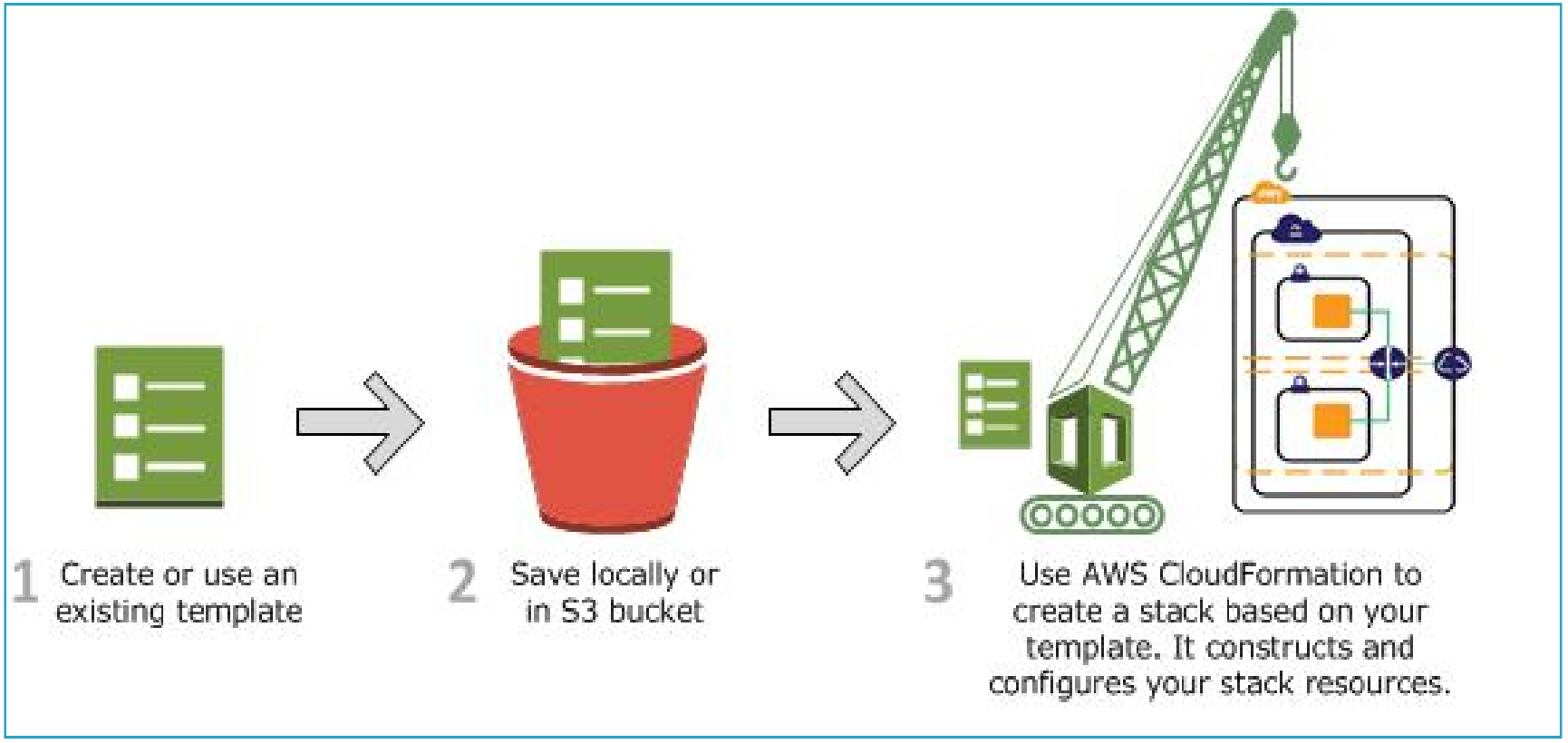
Managed Services

Cloud Formation

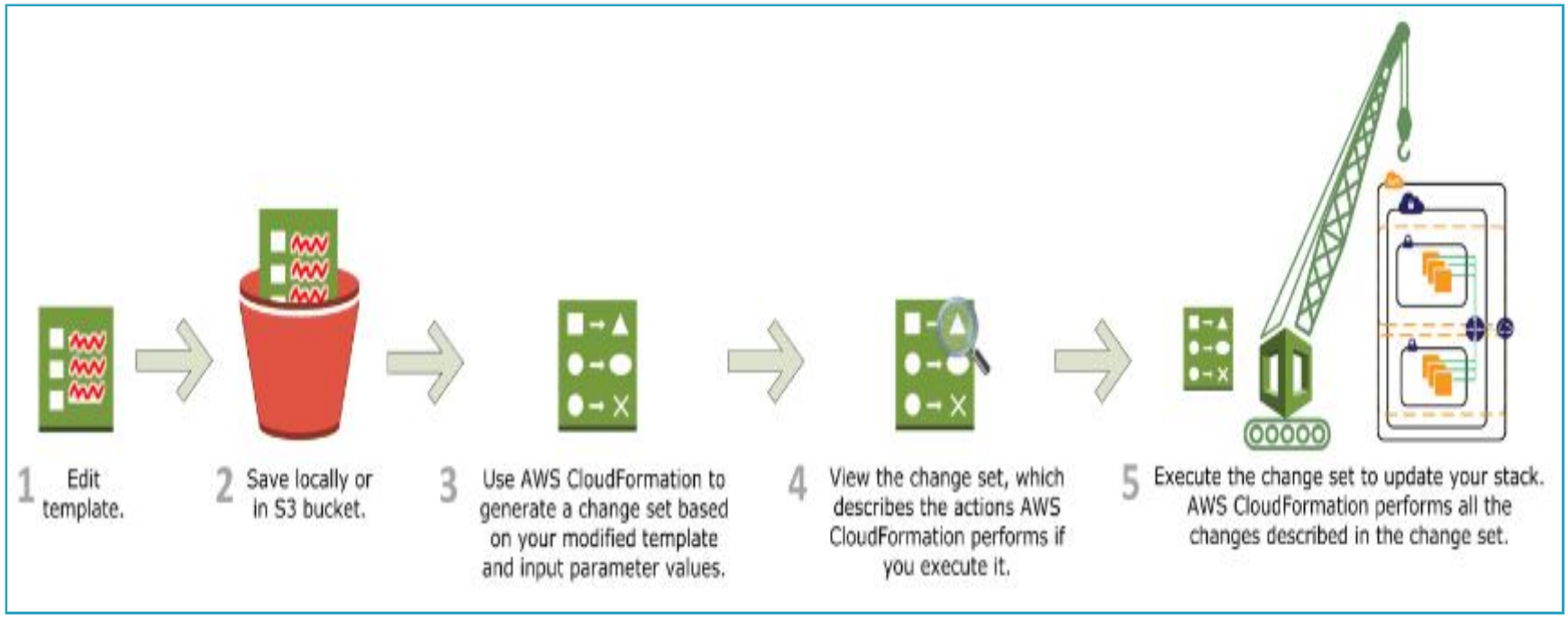
AWS CloudFormation enables you to create and provision AWS infrastructure deployments predictably and repeatedly

- **Templates** – Blueprints for building AWS resources (JSON or YAML)
- **Stack** – Logical group of related resources to be managed as single unit
- **Change Sets** – Shows how changes might impact the running resources

Cloud Formation – Create Process



Cloud Formation – Update Process



Cloud Formation – Best Practices

Planning and organizing

- Organize Your Stacks By Lifecycle and Ownership
- Use Cross-Stack References to Export Shared Resources
- Use IAM to Control Access
- Reuse Templates to Replicate Stacks in Multiple Environments
- Verify Quotas for All Resource Types
- Use Nested Stacks to Reuse Common Template Patterns

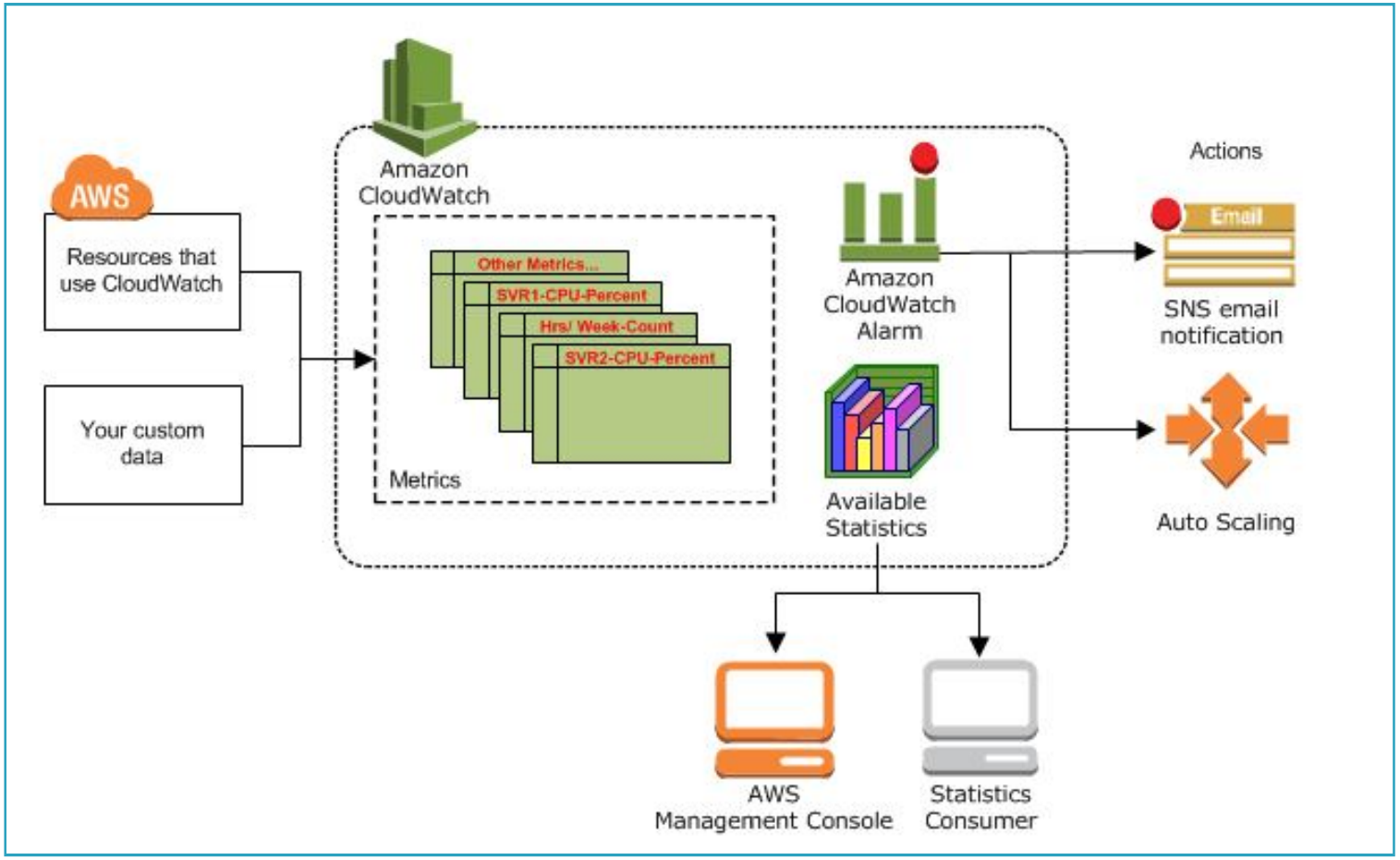
Creating templates

- Do Not Embed Credentials in Your Templates
- Use AWS-Specific Parameter Types
- Use Parameter Constraints
- Use AWS::CloudFormation::Init to Deploy Software Applications on Amazon EC2 Instances
- Use the Latest Helper Scripts
- Validate Templates Before Using Them

Managing stacks

- Manage All Stack Resources Through AWS CloudFormation
- Create Change Sets Before Updating Your Stacks
- Use Stack Policies
- Use AWS CloudTrail to Log AWS CloudFormation Calls
- Use Code Reviews and Revision Controls to Manage Your Templates
- Update Your Amazon EC2 Linux Instances Regularly

Cloud Watch


Amazon
CloudWatch
AWS
Resources that
use CloudWatch
Your custom
data
Other Metrics...
SVR1-CPU-Percent
Hrs/ Week-Count
SVR2-CPU-Percent
Metrics
Amazon
CloudWatch
Alarm
Available
Statistics
Actions
Email
SNS email
notification
Auto Scaling
AWS
Management Console
Statistics
Consumer

AWS Other Services

Analytics

- Athena
- EMR
- CloudSearch
- Elasticsearch Service
- Kinesis
- Data Pipeline
- QuickSight
- AWS Glue

Artificial Intelligence

- Lex
- Amazon Polly
- Rekognition
- Machine Learning

Application Services

- Step Functions
- SWF
- API Gateway
- Elastic Transcoder

Internet Of Things

- AWS IoT
- AWS Greengrass

Business Productivity

- WorkDocs
- WorkMail
- Amazon Chime

Mobile Services

- Mobile Hub
- Cognito
- Device Farm
- Mobile Analytics
- Pinpoint

Game Development

- Amazon GameLift

Contact Center

- Amazon Connect

Thank You!