Amazon ECS, EKS & App Mesh on Outposts

Raja Jadeja
Sr. Product Manager – Technical, Containers

aws

What’s AWS Outposts?

aws

Customer challenges with on-premises applications

IT Infrastructure

Complex procurement and provisioning cycles across a 6–12 vendors and ~3–6 months to get servers installed on-premises

Significant overhead to patch and upgrade on-premises infrastructure against a complex ‘compatibility matrix’ across various hardware and software components

Application maintenance downtime to safely upgrade impacts business continuity and operations

Developers

Don’t have the same services and APIs to build applications on-premises as in the cloud

Don’t have the same tools for automation, deployment, and security controls as in the cloud

Different code and processes for on-premises and cloud applications creates friction and operational risk

Business

Pace of Innovation on-premises lags that in the cloud

AWS Outposts: Bringing AWS on-premises

Same reliable, secure, and high performance infrastructure

Same operational consistency

Same services and APIs

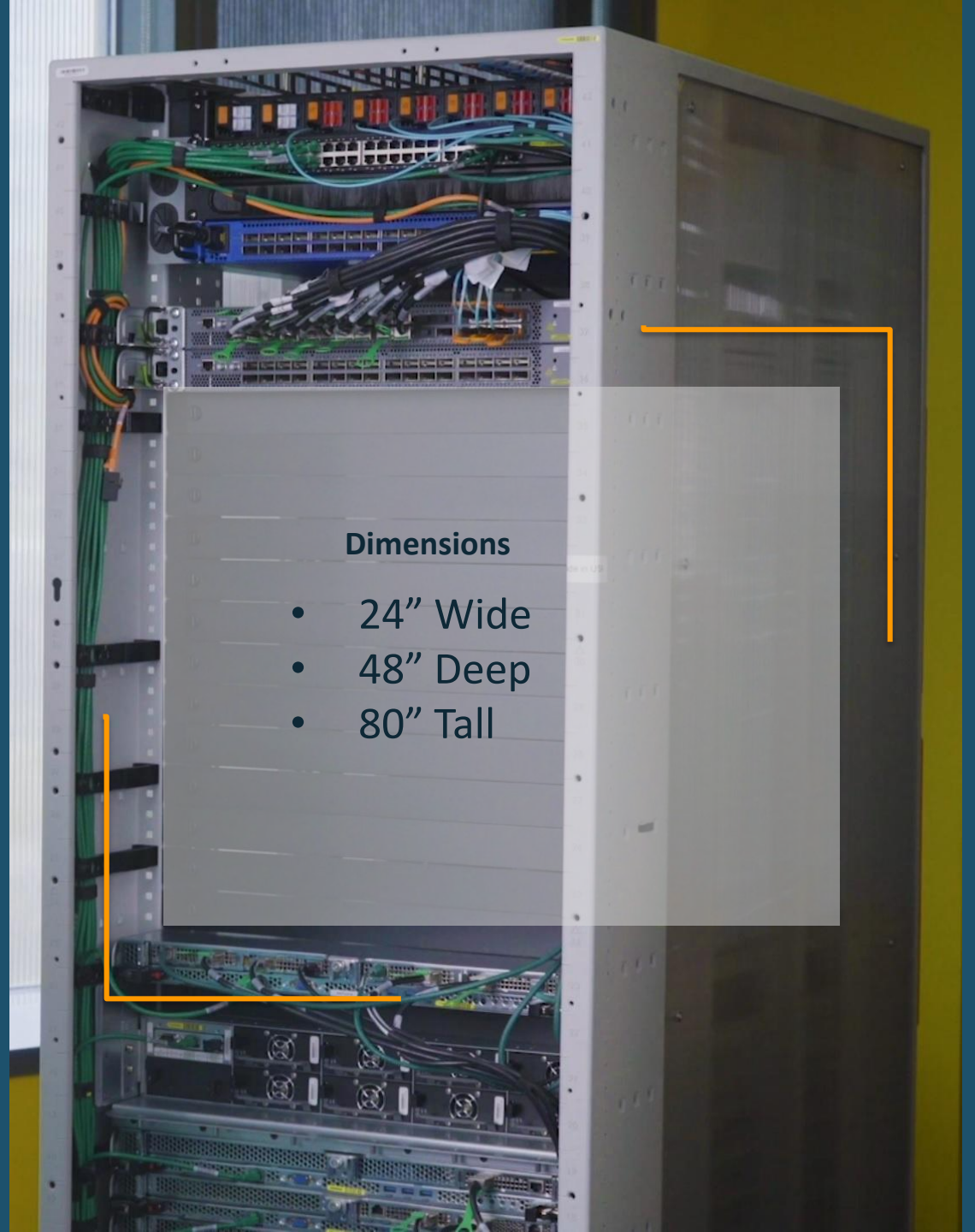
Same tools for automation, deployments, and security controls

Same pace of innovation as in the cloud

aws

AWS Outposts Rack

- Industry standard 42U rack
- Fully assembled, ready to be rolled into final position
- Installed by AWS, simply plugged into power and network
- Centralized redundant power conversion unit and DC distribution system for higher reliability, energy efficiency, easier serviceability
- Redundant active components including top of rack switches and hot spare hosts

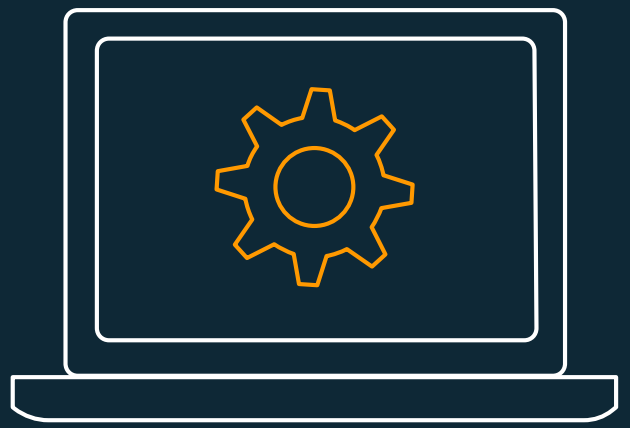


Run AWS services locally

Available at GA or soon after

- Compute & Storage—Amazon EC2 instances and EBS volumes
- Networking—Amazon VPC
- Database—Amazon Relational Database Service (RDS)
- Containers—Amazon Elastic Container Service (ECS), Amazon Elastic Kubernetes Service (EKS), & AWS App Mesh
- Data Processing—Amazon Elastic Map Reduce (EMR)

With the same AWS APIs & tools as in the AWS Region



EC2 Auto Scaling Groups

AWS CloudFormation

CloudWatch

CloudTrail

Elastic BeanStalk

Cloud9

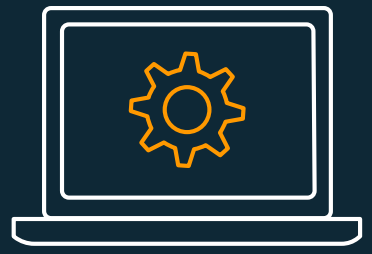
and more…

Containers on Outposts

aws

Applications that need to remain on-premises (fill from next slide)

Applications that are sensitive to latency and variability in latency



- Need for near real time responses to end user applications
- Need to control on-site equipment
- Need to communicate with other on-premises systems

Applications that process data locally

- Need to ensure integrity of ingested signal (e.g. at live events before broadcasting)
- Need to reliably process messages from industrial equipment to monitor production
- Need for managing local data stores

aws

Customer requirements for Containers on Outposts

Requirements:

Simplified stack without having to worry about management, licensing and multiple layers

Desire the same consistent high performance on-premises as AWS Cloud

Single API framework and operation model across cloud and customer locations

Need for data layer (e.g. EBS & RDS) next to containers to achieve desired velocity

Use Cases

Requirements for applications core to competitive advantage to be on-premises

Contractual obligations to vendors and partners

Application modernization challenges – teams with varying capabilities or apps with dependency on legacy architecture

aws

AWS EKS, ECS & AppMesh for Outposts Mindset

- Assume everything will just work … unless noted otherwise
- Of course within the boundaries of a limited set of resources you have (i.e. no spot, no illusion of infinite capacity, etc)
- Control Plane In-Region and Data Plane on Outposts
- Cluster Creation will be different
- Default Load Balancer Choices

aws

AWS EKS, ECS, & AppMesh for Outposts Overview

Most functionality and behavior will be the same with service in the region

Architecture:

- In-region: ECS, App Mesh & EKS Control Plane (Kubernetes Master)
- On Outposts: ECS Container Instances, EKS Worker Nodes, App Mesh Proxy

Traffic Flow to AWS Region:

- Control Plane Traffic: ECS, EKS (master/etcd), App Mesh Config, metrics, CloudWatch, IAM and EC2 API calls
- Application Traffic: Isolate Application traffic to Outposts and local network
- AMIs: ECR in Region -> Cache image locally

aws

AWS EKS & ECS for Outposts Overview

Most functionality and behavior will be the same with service in the region

Load Balancers

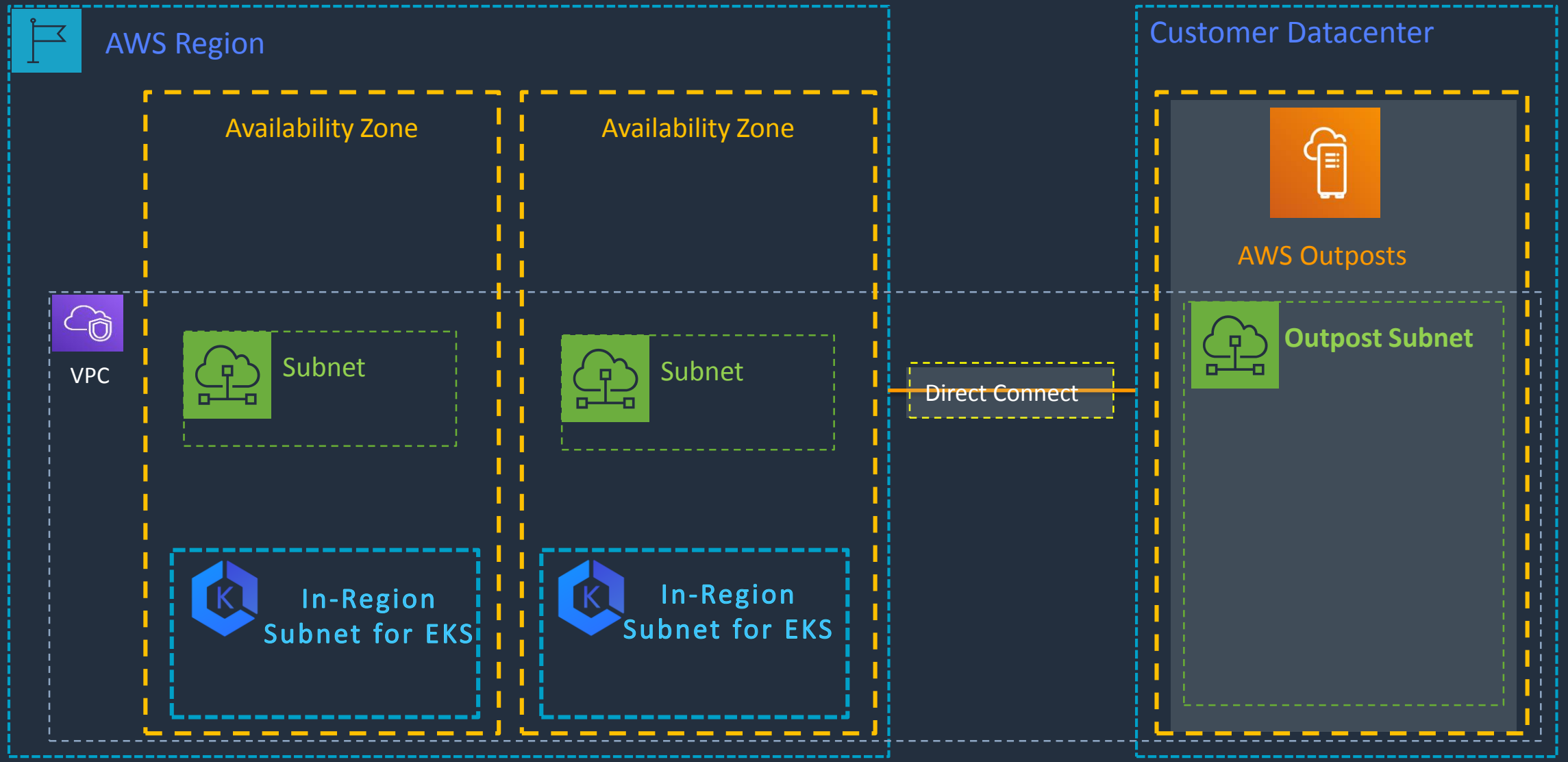
- ALB will be the only local AWS Load Balancer (H2)
- NLB/CLB will be in region but Outposts targets will be supported (incl. for ALB at launch)
- On-premises LBs (F5 BIG IP, Citrix NetScaler) can be used via local gateway

aws

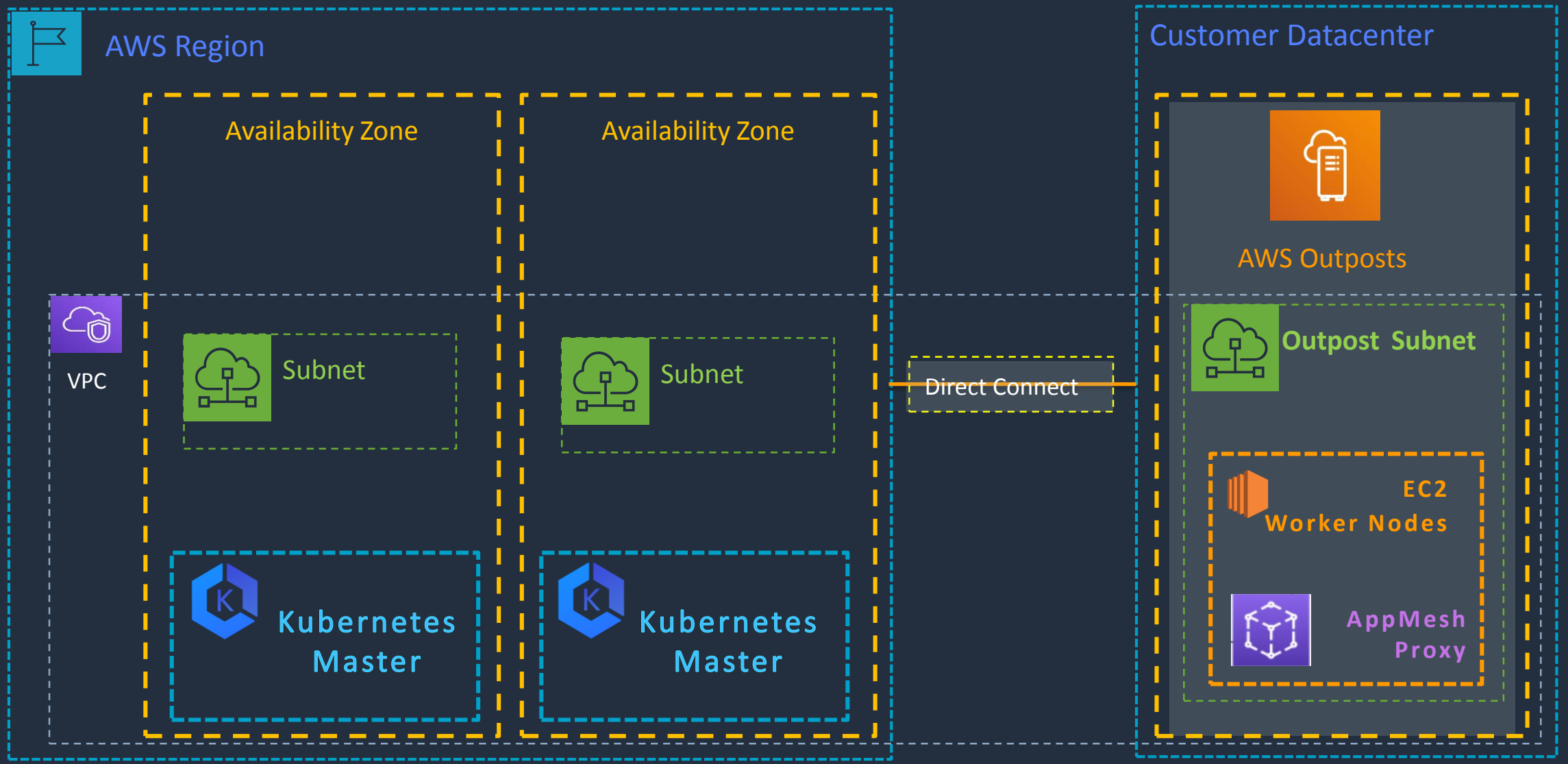
How does this work?

aws

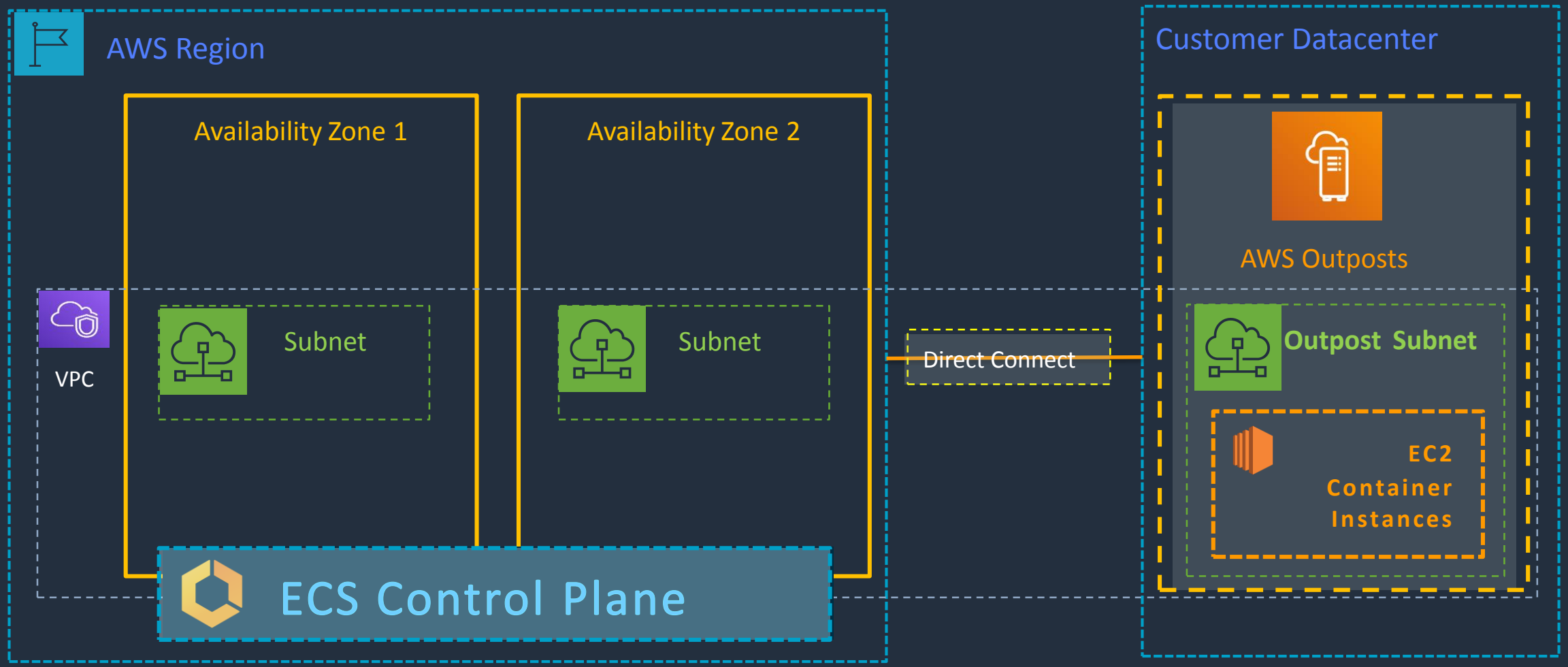
EKS Control Plane & Data Plane Architecture



EKS Control Plane & Data Plane Architecture



ECS Control Plane & Data Plane Architecture



ECS, EKS & AppMesh on Outposts

Amazon ECS

- Subnet needs to be created on Outposts for EC2 instances to populate the cluster with instances.
- Outposts IDs can be used for Placement

Amazon EKS

- Two in-Region subnets for Kubernetes master and Outposts subnet for worker nodes
- Outposts subnet cannot be used for kubernetes master
- CoreDNS can be configured for local DNS

AWS App Mesh

- Setup and configuration is the same as in-region deployments; App Mesh Envoy Proxy deployed on Outposts

aws

Where’s Fargate?

Key Considerations for Customers

1. What is the basis for your desire to run containers on Outposts? Modernization ramp, latency to dependency or locality?
2. Will your clusters stretch between Outposts and in-Region?
3. Do you have any on-premises dependencies? What are the requirements/limitations?
4. How will you integrate/configure networking for Outposts deployments?
5. What are your load balancing requirements and service mesh plans?
6. Do you already have some form of containerization on-premises you are migrating from?

aws

Q & A