Making Kubernetes Useful from Scratch

We're going to instrument kubernetes with the fundamental capabilities that **app teams need to deliver working software to users**.

My hope for your takeaways:

1. You begin to understand how your platform team can iterate with intention in service of developer outcomes
2. You begin to understand the number of decisions required to instrument kubernetes from scratch (spoiler: *a lot*)

Note: This presentation covers CKAD-level kubernetes topics

Every organization with in-house software development has a **platform**, and whether or not they know it, it's a **product** they are providing to **users** (developers).

Some platforms have **better UX** and facilitate better productivity than others.

What Is Tanzu Kubernetes Grid?

Tanzu Kubernetes Grid allows you to make Kubernetes available to developers as a utility, just like an electricity grid.

Operators and developers can use this grid to **create and manage clusters** in the declarative manner that is familiar to Kubernetes users and **keep them version-compatible with upstream Kubernetes**.

Backlog Story

As a Product Manager
I want to access my teams' "demo" application from the public internet
so that I can perform acceptance on features.

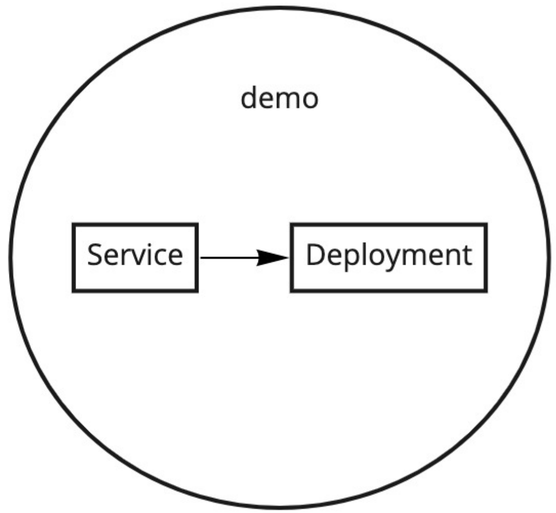
Acceptance Criteria

1. Browsing to https://demo.tkg.samsanders.dev/ presents the 0.0.1 tag of the "demo" application
2. No certificate errors are presented

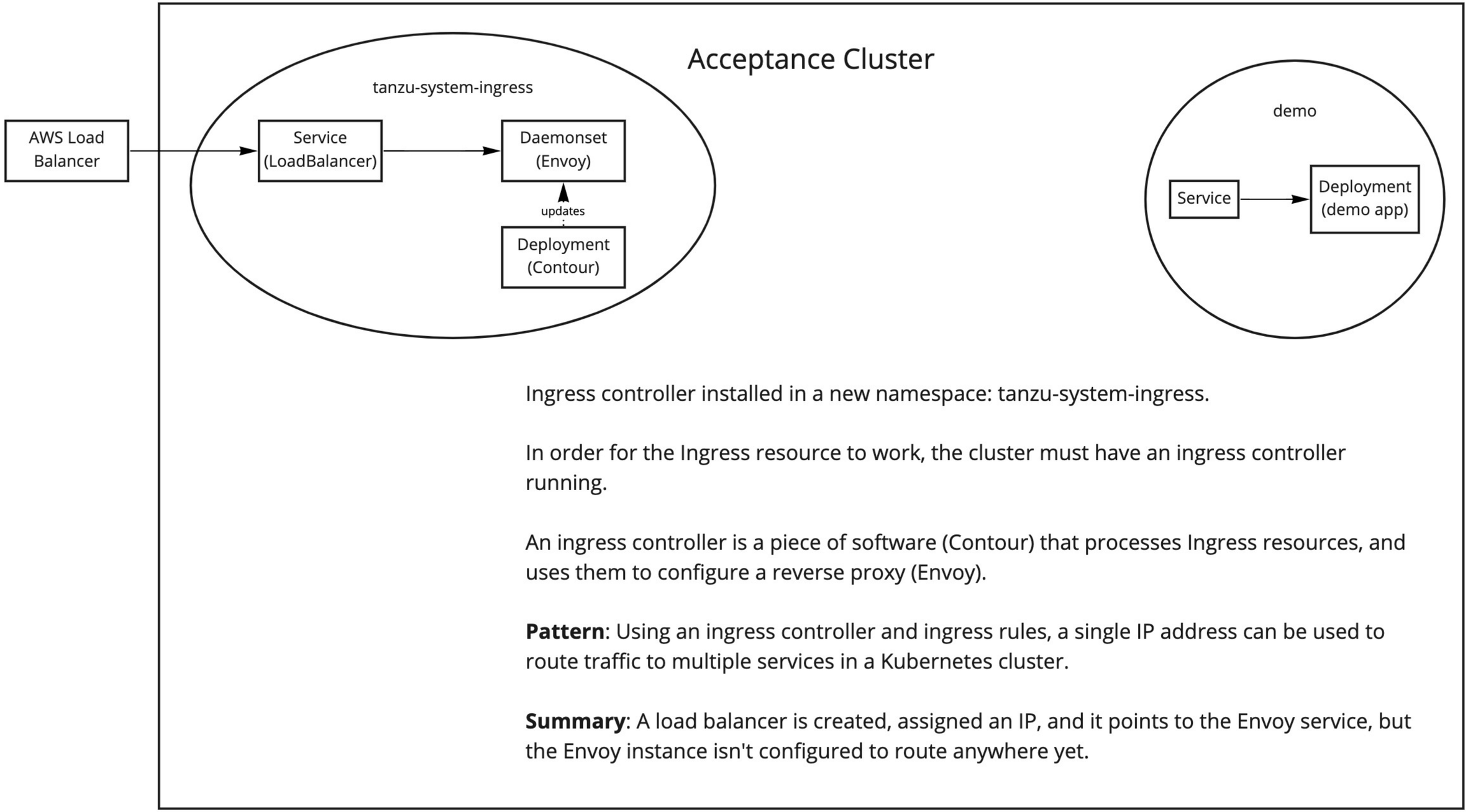
```
→  tkg-platform-demo git:(main) ✗ tanzu cluster create demo-acceptance-workload --file platform/demo-acceptance-workload.yaml
Validating configuration...
Warning: Pinniped configuration not found; Authentication via Pinniped will not be set up in this cluster. If you wish to set up Pinniped after the cluster is created, please refer to the documentat
ion.
unable to override node size
creating workload cluster 'demo-acceptance-workload'...
waiting for cluster to be initialized...
[cluster control plane is still being initialized: , cluster infrastructure is still being provisioned: ]
[cluster control plane is still being initialized: WaitingForControlPlane, cluster infrastructure is still being provisioned: WaitingForControlPlane]
cluster control plane is still being initialized: WaitingForControlPlane
cluster control plane is still being initialized: WaitingForKubeadmInit
cluster control plane is still being initialized: InstanceProvisionStarted @ Machine/demo-acceptance-workload-control-plane-t5jzm
cluster control plane is still being initialized: WaitingForKubeadmInit
waiting for cluster nodes to be available...
waiting for addons installation...
waiting for packages to be up and running...

Workload cluster 'demo-acceptance-workload' created
```

Acceptance Cluster

Acceptance Cluster



deployment with a service in a namespace in a cluster. You can port-forward, but that's all.



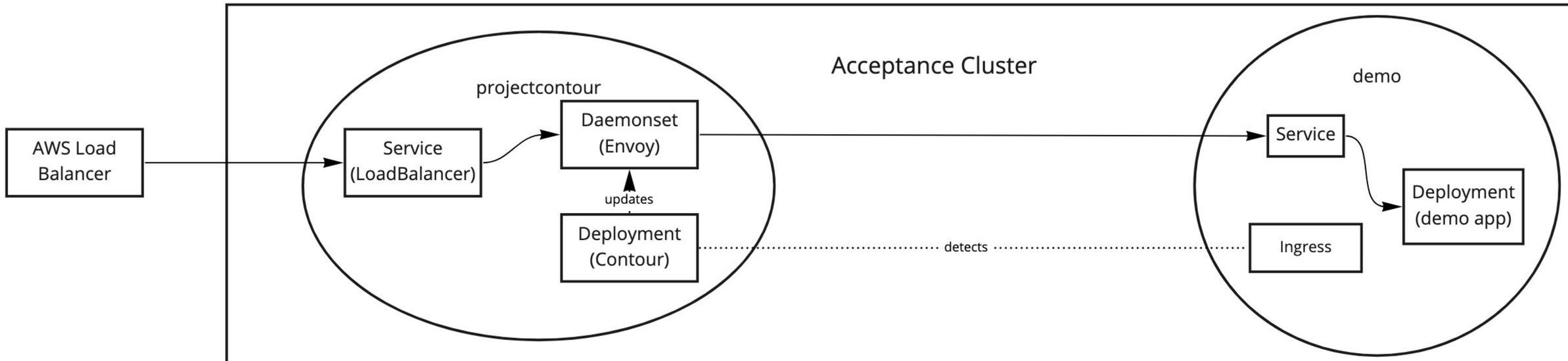
Ingress controller installed in a new namespace: tanzu-system-ingress.

In order for the Ingress resource to work, the cluster must have an ingress controller running.

An ingress controller is a piece of software (Contour) that processes Ingress resources, and uses them to configure a reverse proxy (Envoy).

Pattern: Using an ingress controller and ingress rules, a single IP address can be used to route traffic to multiple services in a Kubernetes cluster.

Summary: A load balancer is created, assigned an IP, and it points to the Envoy service, but the Envoy instance isn't configured to route anywhere yet.



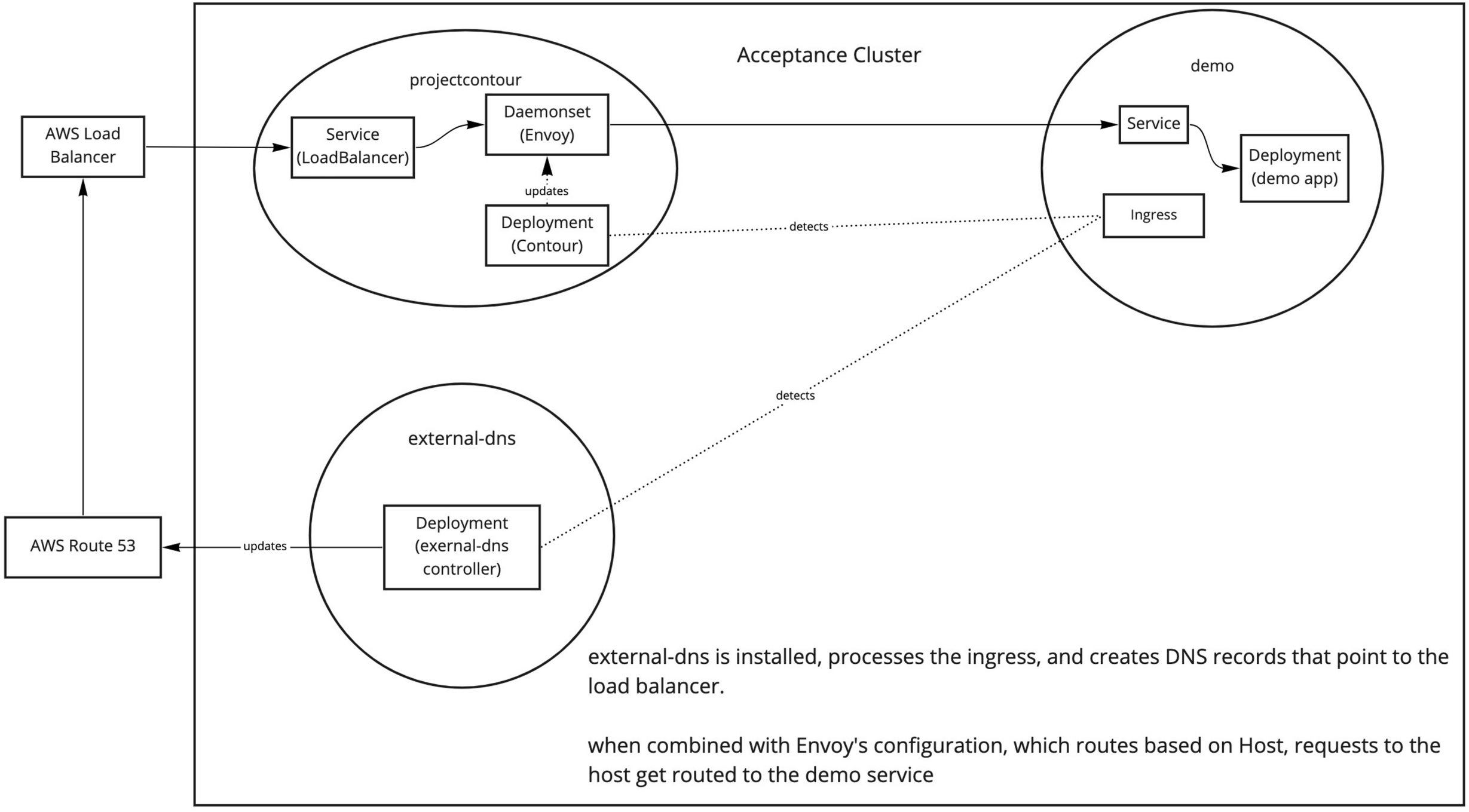
"Traffic routing is controlled by rules defined on the Ingress resource."

https://kubernetes.io/docs/concepts/services-networking/ingress/#what-is-ingress

Contour processes Ingresses and updates Envoy accordingly (configuration).

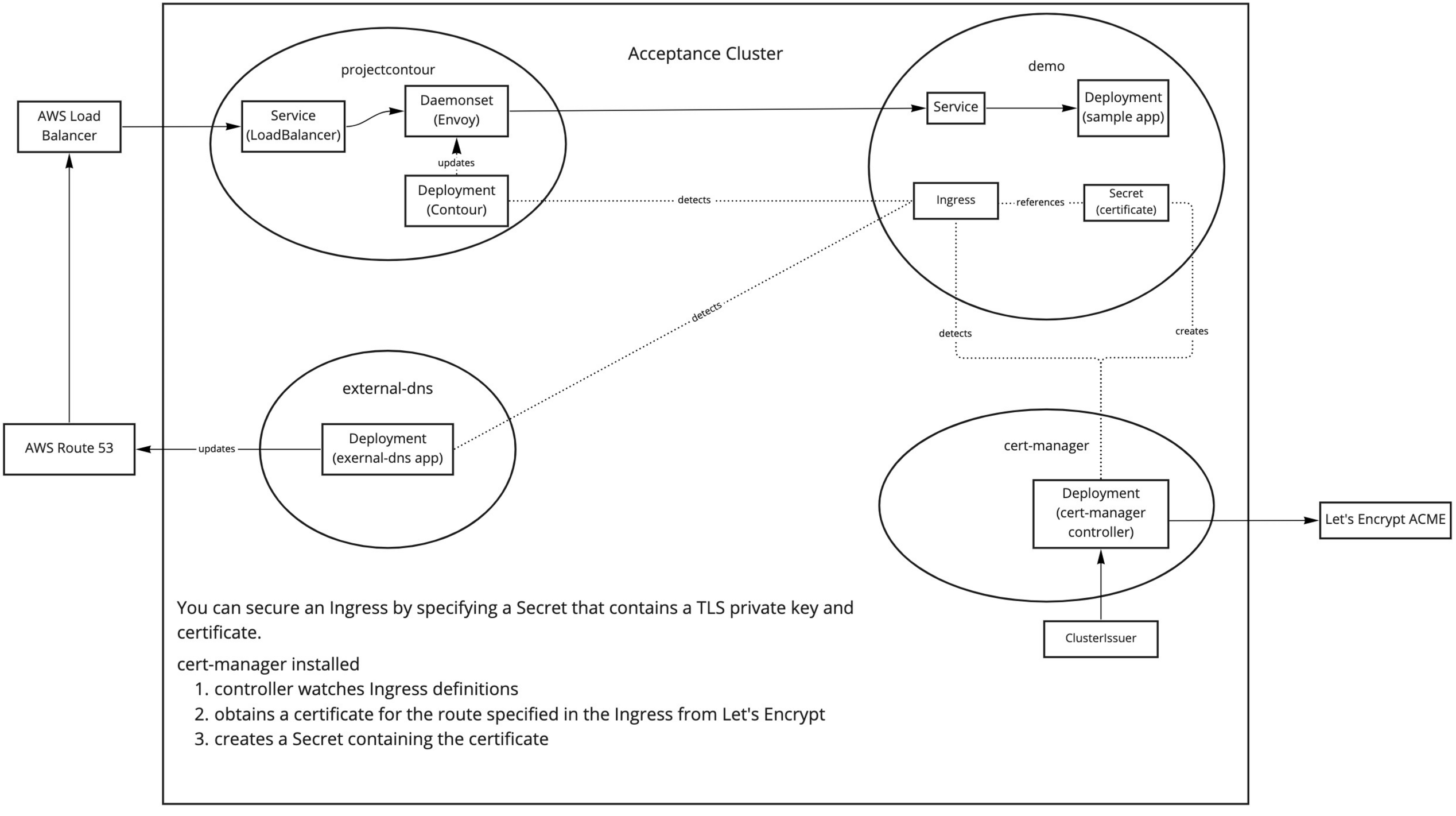
Host header based routing: requests to the load balancer with the Host header set to the host specified in the demo Ingress will now route to the demo service.

Notes:
"The "Host" header field in a request provides the host and port information from the target URI, enabling the origin server to distinguish among resources while servicing requests for multiple host names on a single IP address."
https://www.rfc-editor.org/rfc/rfc7230#section-5.4
(Good routing strategy because the proxy is unknown to the client -- they make requests to a hostname mapped to an ip address, so many can map to one, and the proxy can intercept to route.)



external-dns is installed, processes the ingress, and creates DNS records that point to the load balancer.

when combined with Envoy's configuration, which routes based on Host, requests to the host get routed to the demo service



You can secure an Ingress by specifying a Secret that contains a TLS private key and certificate.

cert-manager installed

1. controller watches Ingress definitions
2. obtains a certificate for the route specified in the Ingress from Let's Encrypt
3. creates a Secret containing the certificate

Q&A