Meetup: Containers in AWS

Overview by Chris Amatangelo

Speaker: Jason Poley, Cloud Architect at Macquarie Group

Meetup:

https://www.meetup.com/gpawsug/events/276115593/

Event summary:

Cloud architect Jason Poley gave a high-level overview of the current state of container related service offerings on AWS - where we are, and where we are going.

Why is this relevant to the class?

There is an increasing movement towards being cloud-native, using containers and utilizing serverless architectures where, rather than provisioning and configuring your virtualized hardware, your cloud provider does all infrastructure management for you.

While the world of container orchestration and expertise is a specalty in and of itself, as system administrators it's important to have an understanding of the current capabilities/trends surrounding the technology that touches our domain. As containerized and cloud-provider-managed infrastructure continues to become an increasingly popular approach, I believe that a number of traditional system administration roles will organically morph closer and closer to roles akin to devops engineers.

Why did I chose this meetup?

I chose to attend this meetup because of my increasing interest (as well as the increasing relevance) of container technology on AWS and other cloud providers. Additionally playing off of the article that I presented in class around primitive concepts that underlie containiers (i.e. namespaces, cgroups and UFS), it's both fascinating and important to learn about the multitude of companies, products and services that provide novel features on top of the core concept of a container.

Additionally, given that AWS is the land of a thousand different ways to do roughly the same thing, I wanted to get a sense of what products and services existed that I was unaware of and how they differ from ones that either, I was familiar with, or had used in the past. Like all big tech companies, AWS has purchased numerous companies that focus entirely on expanding the capabilities and strengths of container technology.

What did I learn?

Rather than simply summarizing high-level descriptions of each container-based service presented during the meetup, let's instead imagine that we are in an organization that makes heavy use of containers and we are in charge of investigating and deciding how utilizing what AWS has to offer might streamline or improve our current process.

As admins, we care about scalability, reliability, manageablility, security and performance - and while stepping into container land is like opening pandora's box, which can leave your head spinning, each product can more-or-less be boiled down to a single concept. To make these product and service descriptions concise and a little more digestable let's ask ourselves two questions that could drive our decision making process - do we want orchestration, and do we have a particular focus on security?

First, do we want orchestration?

We know that we run our software using a number of containers, so maybe we would like to automate the provisioning, scaling, deployment, etc., of our infrastructure by making use of container orchestration techonology? Maybe we want to use Kubernetes, arguably one of the most hyped-up technologies of the last 6-7 years, or some other orchestration engine? Maybe we want to switch from treating our infrastructure as special "snowflakes" that we ssh into but rather as cattle that we can instantly spin up and tear down - or maybe we like control and are content with the way our infrastructure is managed? So how might we answer this question as an informed customer of AWS.

"Yes - we want Kubernetes but we want to provision and configure everything entirely ourselves - we need maximum flexibility so we will roll our own Kubernetes installation on EC2, this will certainly keep us busy."

"Yes - we want Kubernetes but we want AWS to manage the orchestration environment, but we need to stick to the open-source Kubernetes jargon, APIs and eco-system - no vendor lock-in allowed here so let's make use of EKS"

"Yes - we care most about convenience - we want AWS to manage the orchestration environment and we are happy to use the AWS flavor of Kubernetes - let's go ahead and use ECS"

"Yes - we are drinking the kool-aid and want AWS to manage the orchestration engine and we want to be serverless for the hosts as well - we will use Fargate on either ECS or EKS."

"Yes - but we have very specific latency requirements, so we are considering AWS Outposts in combination with either ECS or EKS."

"Yes - but we want to leverage AWS tooling but we need the cluster to be on-prem, so let's wait for EKS Anywhere to be released."

"No - we don't need orchestration and we don't want to manage any infrastructure - we want to go cutting-edge and use Lambda Containers"

Or do we have a specific focus on security regardless of our orchestration needs?

"Yes, let's use that new OS written in Rust that is purpose-built as a Kubernetes node - Bottlerocket, after Microsoft says 70% of their vulnerabilites are memory corruption bugs!

"Yes and we want really want enclaves so let's consider Nitro.

"Yes and we are interested in the light-weight VMs that are super snappy and can start in 200ms - let's see what Firecracker has to offer.

While AWS and other cloud-providers will obviously continue to allow granular provisioning, configuration and management to its customers for the forseeable future, it's worth acknowledging and focusing attention on the evergrowing number of new products that center around serverless architecture and containers. Whether it be entirely new operating systems with a focus on a containerized setup or increasingly customizable methods of leveraging container orchestration engines, containers may just be getting started.