

Azure AKS Accelerator

Azure Kubernetes Service accelerator brings your organization from zero to deployed apps, policies, and security in seconds

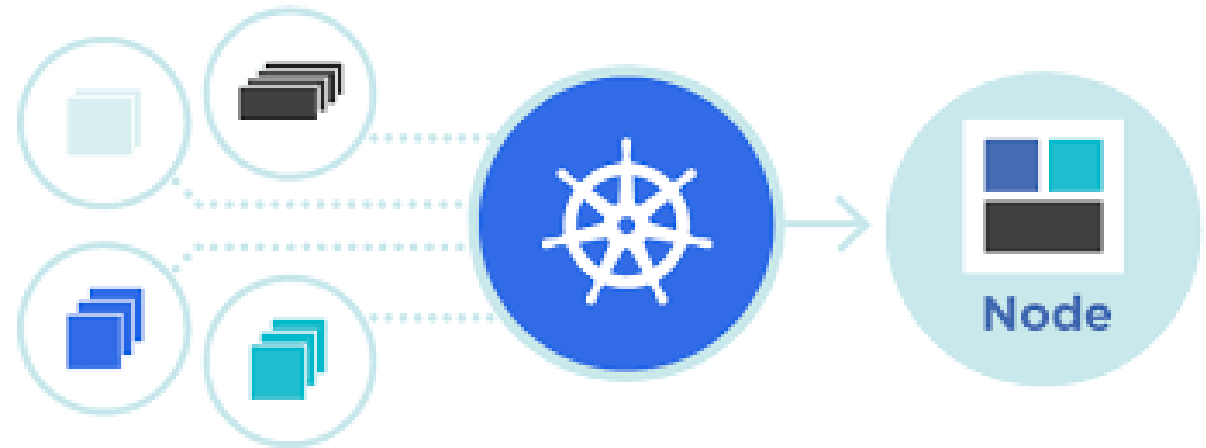
**Containers reduce environment drift.
Container orchestrators provide availability,
simplify deployment and maximize
infrastructure utilization.**

Kubernetes

“By 2022, more than 75% of global organizations will be running containerized applications in production”

– Gartner

Kubernetes is the leading container orchestrator.



Overview

The Slalom AKS Accelerator was created in order to shorten the onboarding, or Sprint 0, of all projects using AKS.

The accelerator functions from Slalom's 4 core principles.

Core principles

Modular

Each section can be [separately deployed](#) and supports separation of concerns

Secure

Introducing tools like Checkov, and SonarQube help to make sure our [code is secure](#) from IaC to app code and as far "left" in the deployment cycle for [faster debugging](#) feedback

End-to-end

The accelerator handles the [set up](#) for the underlying infrastructure, and [the KEY connections](#) between the Azure services and the application

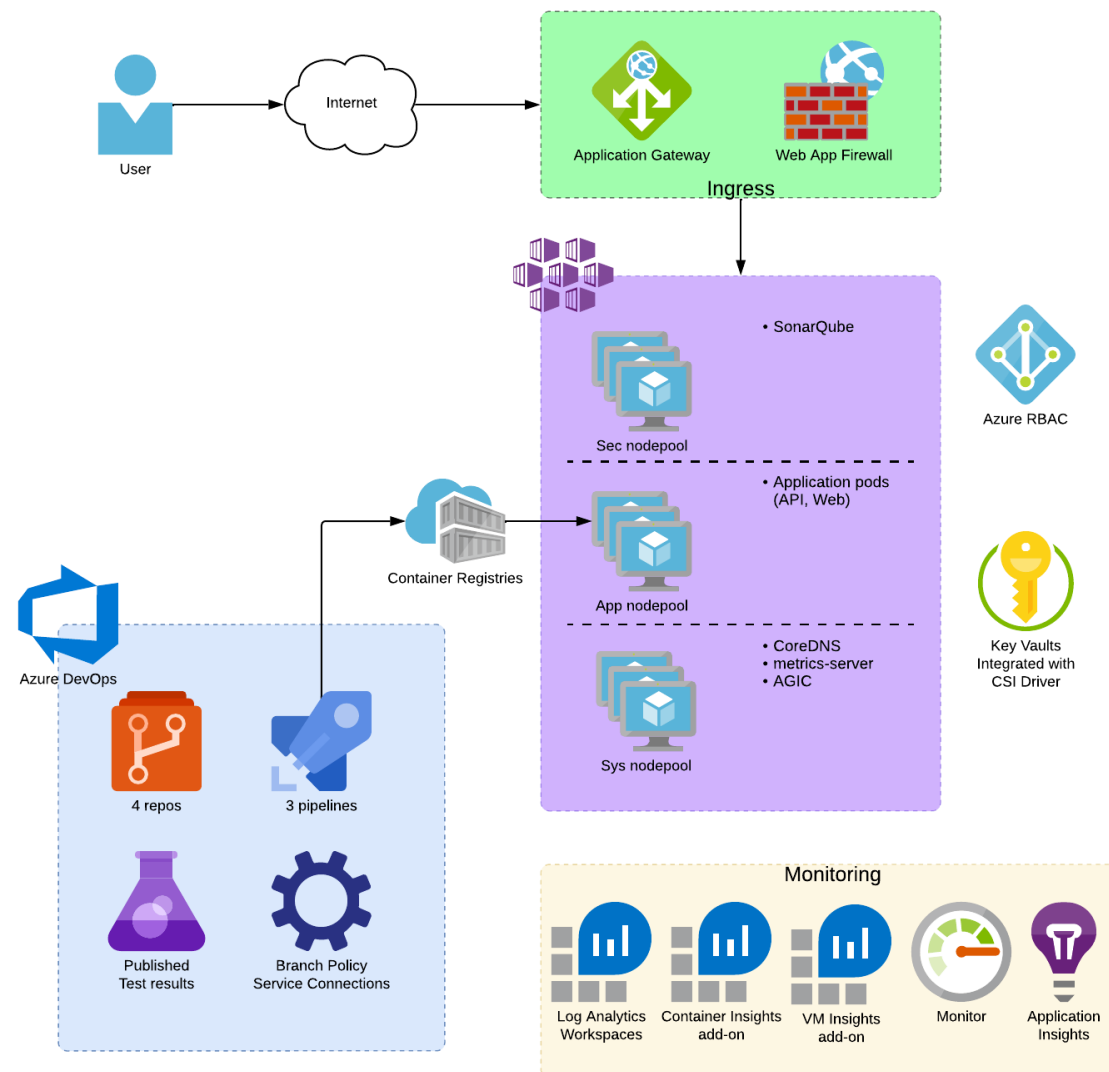
MVP

We start with a secure minimum viable product and customize it to fit your organization and your unique needs for [flexible scaling](#) and [high-end compute](#), all while [minimizing deployment risk](#)

User Flow & High-Level Architecture

Key features of the Slalom Architecture:

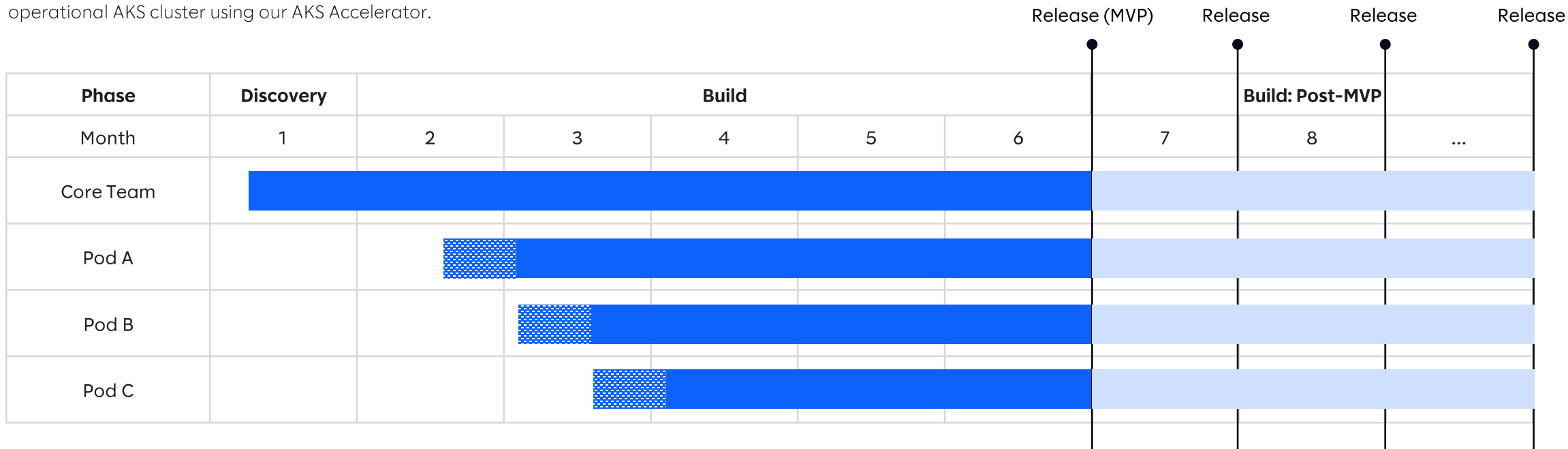
- Rapid transition from OpenShift to AKS- deployment of containerized applications and hosting for included clusters
- Connections from Azure DevOps to the container registry, and to AKS to deploy applications - a reusable and quickly customized code base enables rapid deployment for all 3 pipelines
- Application Lifecycle Management (ALM) best practices – after many successful deployments, Slalom has consolidated the best practices for ALM and product development within this accelerator
- Monitoring – The Slalom AKS Accelerator uses templates for all aspects of monitoring on Azure, massively decreasing time and cost for development



Timeline

AKS Accelerator

This is a rough estimated timeline to go from nothing to an operational AKS cluster using our AKS Accelerator.



Why use Containers?

Containers enable you to build modern applications or package existing applications to increased portability and reduce environment drift. Container orchestrators allow you distribute your containers intelligently across your infrastructure, either in the cloud or on-premises. Furthermore, container orchestrators simplify deployment processes, ensure high availability, provide scaling capabilities, and may abstract capabilities from your application. This reduced overhead lets developers reclaim time and energy that can be spent on developing great products which scale and are reliable.

Maximize Compute Infrastructure

Container orchestrators intelligently schedule containers on machines to maximize the utilization of the host.

Flexible Scaling

Your application can be scaled automatically or by adjusting its capacity through toggling the units of consumption (e.g. throughput, memory) rather than units of individual servers.

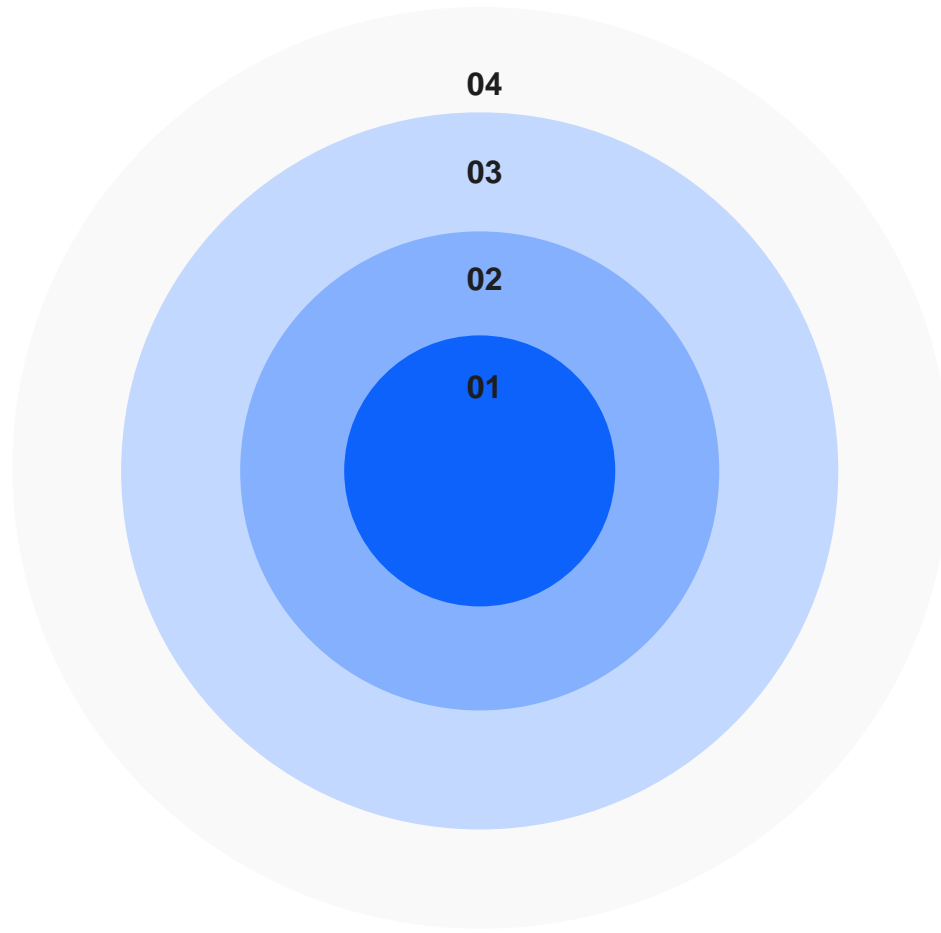
Automated High Availability

Container orchestrators provide high availability. You don't need to architect for these capabilities, if a node or host dies, your containers will automatically be rescheduled.

Lower Risk Deployments

Not only do containers ensure that your code will work the same way in every environment, but container orchestrators allow for rolling deployments.

Kubernetes Technical Concepts



01 Pod

A group of one or more tightly coupled application containers that you scale and schedule together.

02 Deployment

Defines the number of particular pods to live in the cluster. Provides self-healing and high availability

03 Service & Ingress

Services expose your pods or deployments as an addressable service. Ingresses allow you to access your services outside your Kubernetes cluster

04 Namespace

Allows you to logically partition your Kubernetes cluster