3: Scalability Framework

WHY THIS FRAMEWORK IS IMPORTANT

The goal of Kubernetes is to scale three ways:

1. Does your infrastructure scale reliably and securely?
2. Does your Kubernetes environment autoscale?
3. Do your Kubernetes applications self-heal?

This framework focuses on understanding the temporal requirements of applications using your environment.

EXERCISE


1 Hours


Easy Effort


Enterprise Architect
Head of Development
Security Manager

Are Kubernetes requests and limits used with every deployment within a cluster?

- ☐ Yes
- ☐ No

Are node autoscalers currently in use?

- ☐ Yes
- ☐ No

Is horizontal pod autoscaling in use?

- ☐ Yes
- ☐ No

Is auto scaling functionality being continually tested?

- ☐ Yes
- ☐ No

Are cluster autoscalers and workloads designed to respond to availability zone outages?

- ☐ Yes
- ☐ No

Are autoscalers leveraging limits and requests?

- ☐ Yes
- ☐ No

What group is responsible for determining requests and limits for a new workload?

- ☐ Operations team
- ☐ Development team
- ☐ DevOps team
- ☐ Other

Is there a mechanism for performing canary style deployments and segmenting traffic?

- ☐ Yes
- ☐ No

NEXT STEPS

Count the number of times you answered “Yes” and compare it to the number of times you answered “No.” This will give you a sense of how prepared your Kubernetes environment is to auto scale.

Make sure you are building a foundation that can scale on demand with ease. This will allow you to expand your Kubernetes capabilities across your organization and deliver rapid application deployment to product owners.