

Provisioning Dedicated Game Server on Kubernetes Cluster

Open Infra Days Korea 2018

김진웅
SK C&C

Profile

김진웅

2006년~2016년 - 삼성SDS

- 삼성관계사 레거시/클라우드 인프라 운영

2017년~현재 - SK주식회사 C&C Cloud Lab

(Big Challenge!)

- From vSphere to OpenStack Migration 플랫폼 개발
- Kubernetes 기반 Deep Learning Cloud(/w GPU) 플랫폼 개발
- Kubernetes 기반 Serverless 플랫폼 개발
- Kubernetes 기반 IaaS Automation 플랫폼 개발
- Game/Media Cloud R&D

Kubernetes



- Kubernetes (k8s)
 - 2014년 Google이 시작한 프로젝트로, 애플리케이션 컨테이너의 배포, 스케일, 오퍼레이팅을 자동화 해주는 오픈 소스 플랫폼
(Orchestrates Computing, Networking, Storage Infrastructure)
 - Objects (v1.10 기준 beta이상)
 - Workloads : Pod, Job, Cronjob(beta), DaemonSet, Deployment, ReplicaSet, StatefulSet
 - Discovery & LB : Endpoint, Service, Ingress(beta)
 - Config : ConfigMap, Secret, PVC, StorageClass, Volume, VolumeAttachment(beta)
 - Metadata : ControllerRevision, CustomResourceDefinition(beta), Event, LimitRange, HorizontalPodAutoscaler, MutatingWebhookConfiguration(beta), ValidatingWebhookConfiguration(beta), PodTemplate, PodDisruptionBudget, PodSecurityPolicy(beta)
 - Object : APIService, Binding, CertificateSigningRequest(beta), ClusterRole, ClusterRoleBinding, ComponentStatus, LocalSubjectAccessReview, Namespace, Node, PersistentVolume, ResourceQuota, Role, RoleBinding, SelfSubjectAccessReview, SelfSubjectRulesReview, ServiceAccount, SubjectAccessReview, TokenReview, NetworkPolicy

CustomResourceDefinition



- k8s Object를 확장해서 사용할 수 있는 가장 간단한 방법
 - Resource 당 하나의 API 버전만 지원
 - 1.8+ 이후부터 JSON 스키마 유효성 검증 가능
 - 동적으로 원하는 resource를 만들수 있음
 - K8s database(etcd)와 API를 그대로 활용할 수 있음
 - CRUD 가능 (Create, Read, Update, Delete)
 - 주의사항 : etcd가 별도 분리된 managed서비스나 etcd 인스턴스가 분리된 환경에서 사용 권고
- 주요 활용용도
 - 애플리케이션 정보 저장
 - RouteRule on istio (Canary Launch)
 - GameServer on Agones

CustomResourceDefinition



- Define CRD

```
1apiVersion: apiextensions.k8s.io/v1beta1
2kind: CustomResourceDefinition
3metadata:
4  name: widgets.example.crd.com
5spec:
6  group: example.crd.com
7  version: v1
8  scope: Namespaced
9  names:
10   plural: widgets
11   kind: Widget
```

```
$ kubectl get CustomResourceDefinition
NAME                                     AGE
fleetallocations.stable.agones.dev     11d
fleets.stable.agones.dev               11d
gameservers.stable.agones.dev          11d
gameserversets.stable.agones.dev       11d
widgets.example.crd.com                2m
```

CustomResourceDefinition

- Check CRD Status

```
$ kubectl get crd widgets.example.crd.com -o yaml
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  creationTimestamp: 2018-06-22T04:02:34Z
  generation: 1
  name: widgets.example.crd.com
  resourceVersion: "396072"
  selfLink: /apis/apiextensions.k8s.io/v1beta1/customresourcedefinitions/widgets.example.crd.com
  uid: 17c85900-75d1-11e8-bb5d-468c0b1e1c32
spec:
  group: example.crd.com
  names:
    kind: Widget
    listKind: WidgetList
    plural: widgets
    singular: widget
  scope: Namespaced
  version: v1
status:
  acceptedNames:
    kind: Widget
    listKind: WidgetList
    plural: widgets
    singular: widget
  conditions:
  - lastTransitionTime: 2018-06-22T04:02:34Z
    message: no conflicts found
    reason: NoConflicts
    status: "True"
    type: NamesAccepted
```

CustomResourceDefinition



- Create CRD

```
apiVersion: example.crd.com/v1
kind: Widget
metadata:
  name: my-widget
location:
  city: seoul
  place: hotel
weather: sunny
```

```
$ kubectl get widget my-widget -o yaml
apiVersion: example.crd.com/v1
kind: Widget
location:
  city: seoul
  place: hotel
metadata:
  clusterName: ""
  creationTimestamp: 2018-06-22T10:54:00Z
  generation: 1
  name: my-widget
  namespace: default
  resourceVersion: "403551"
  selfLink: /apis/example.crd.com/v1/namespaces/default/widgets/my-widget
  uid: 91b528c5-760a-11e8-bb5d-468c0b1e1c32
weather: sunny
```

MutatingAdmissionWebhook



- Admission controller 확장 기능
- Admission Controller - k8s의 api-server의 resource나 object(pod, service등)의 생성 요청을 가로채 제어 할수 있는 강력한 도구(플러그인)
- 클러스터 관리자가 바이너리로 컴파일 하고 구성해야 하기 때문에 유연하게 사용하기 어려움
- 1.7 버전 이후 위 제약사항을 해결하기 위해 alpha feature로 [Initializers](#) and [External Admission Webhooks](#) 기능이 도입됨
- [External Admission Webhooks](#) 는 k8s 리소스의 유효성 검사를 하는데 활용, 유효성 검사를 통과 하지 못하면 해당 리소스는 쿠버네티스에서 생성되지 않게 할 수 있음.
- 1.9 버전 이후 [External Admission Webhooks](#) 은 beta로 승격, [MutatingAdmissionWebhook](#) 및 [ValidatingAdmissionWebhook](#)으로 나뉘었지만 [Initializers](#) 는 alpha 로 유지
- **MutatingAdmissionWebhook** 은 유효성 검사 이외에도 승인 과정시 k8s object에 변경 가능
 - 예를 들면 Object Create 과정에서 resource quota를 변경한다던지 Agones 및 istio와 같이 Sidecar-injection 방식의 Custom Resource를 수정하여 object를 생성이 가능함
 - Webhook 방식은 gRPC 프로토콜을 사용하는 데 개발언어에 구애 받지 않고 확장을 할 수 있다는 장점이 있음

Mutating Admission Webhook



- 언제 쓰는가?
 - Webhook을 사용하여 k8s cluster-admin이 api-server를 다시 컴파일하지 않고도 object생성 요청시 mutating(변경) 및 validation(유효성검증) 을 하는 플러그인을 만들 수 있다
 - 이를 통해 개발자는 모든 resource 에서 여러 작업 (CRUD)에 대한 승인 로직에 대해 사용자 정의 할 수 있는 유연성을 제공함
- Use-Case
 - resource를 생성하기 전에 변경 (예, [Istio](#) 에서 처럼 traffic management 와 policy enforcement 을 위해 [Envoy](#) sidecar container를 injection)
 - StorageClass Provisioning 자동화 (PersistentVolumeClaim object 생성을 모니터링 하고 미리 정의 된 정책에 따라 객체에 storage를 자동으로 추가. 사용자는 StorageClass 생성 에 신경 쓸 필요가 없음)
 - 복잡한 custom resource 검증 (Agones와 같은)
 - 특정 namespace에서 Resource생성 제한 :
멀티 테넌트 시스템에서 reserved namespace에 resource생성을 금지시킬 때

Agones



- 그리스어로 경연, 경기
- 2018년 3월 13일 오픈소스로 공개한 게임 서비스 라이브러리 (구글, Ubisoft)
- FPS, MMO, MOBA_(Multiplayer Online Battle Arena) 같은 멀티플레이 게임을 **k8s** 환경에서 호스팅, 실행, 스케일링하기 위한 라이브러리
- Alpha(0.2.0)
- 주요기능
 - “**GameServer**”라고 하는 k8s **CustomResource**를 API로 컨트롤 가능
 - “GameServer”의 Status, Connection info, Lifecycle 관리
 - 게임 클라이언트의 SDK 통합가능 (**Sidecar**)
- 요구사항
 - k8s Cluster 1.9+
 - Minikube, GKE(Google), AKS(Azure), IKS(IBM) 테스트 완료
 - **MutatingAdmissionWebhook, ValidatingAdmissionWebhook** admission controllers
<https://kubernetes.io/docs/reference/access-authn-authz/admission-controllers/#is-there-a-recommended-set-of-admission-controllers-to-use>
 - 게임서버 UDP 포트 방화벽 오픈
 - 게임서버 클라이언트 SDK 필수 (C++, GO, Rust, REST 지원)

Agones



- Agones 설치

```
$ kubectl create clusterrolebinding cluster-admin-binding \
--clusterrole=cluster-admin --serviceaccount=kube-system:default
```

- YAML 배포 <https://raw.githubusercontent.com/ddiwoong/agones/master/install/yaml/install.yaml>
 - Manifests 요약
 - Namespace, RBAC
 - Deployment : Agones-controller
 - CustomResourceDefinition
 - GameServer
 - GameServerSet
 - Fleet
 - FleetAllocation
 - ValidatingWebhookConfiguration
 - MutatingWebhookConfiguration

- Helm 배포

```
$ git clone https://github.com/GoogleCloudPlatform/agones.git
$ cd install/helm/
$ helm install --name my-release agones
```

Agones GameServer



- GameServer

```
$ kubectl apply -f
https://raw.githubusercontent.com/GoogleCloudPlatform/agones/master/examples/simple-
udp/server/gameserver.yaml
gameserver.stable.agones.dev "simple-udp" created

$ kubectl get gameservers
NAME          AGE
simple-udp     7s

$ kubectl get pods
NAME          READY    STATUS    RESTARTS   AGE
simple-udp-qms6f 2/2      Running   0          13s
```

Agones GameServer

- GameServer 상태확인

```
$ kubectl describe gameserver simple-udp
...#생략
API Version:  stable.agones.dev/v1alpha1
Kind:          GameServer
Metadata:
...#생략
Spec:
  Container:      simple-udp
  Container Port:  7654
  Health:
    Failure Threshold:  3
    Initial Delay Seconds: 5
    Period Seconds:     5
  Host Port:      7787
  Port Policy:    dynamic
  Protocol:       UDP
  Template:
    Metadata:
      Creation Timestamp: <nil>
    Spec:
      Containers:
        Image:  gcr.io/agones-images/udp-server:0.1
        Name:   simple-udp
      Resources:
Status:
  Address:  169.56.94.52
  Node Name: 10.178.188.16
  Port:      7787
  State:     Ready
Events:
  Type      Reason          Age   From                  Message
  ----      -
  Normal    PortAllocation  40s   gameserver-controller Port allocated
  Normal    Creating        40s   gameserver-controller Pod simple-udp-qms6f created
  Normal    Starting        40s   gameserver-controller Synced
  Normal    Ready           26s   gameserver-controller Address and Port populated
```

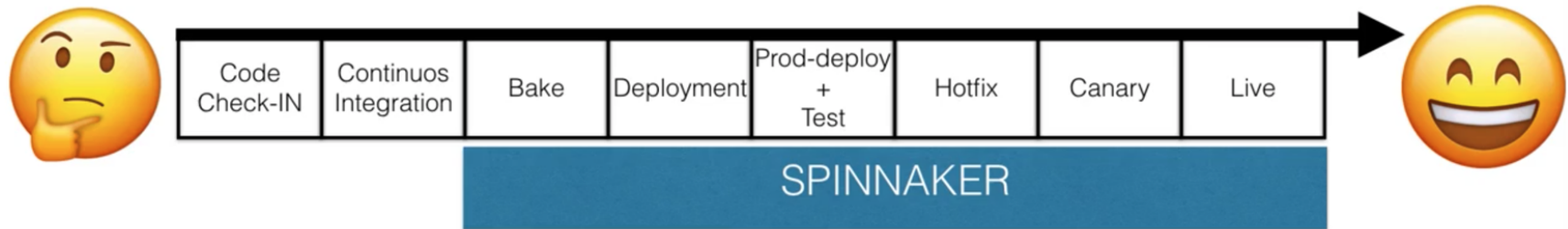
Agones GameServerFleet



- Agones GameServerFleet
: warm GameServer Fleet 을 생성하고 게임플레이시 마다 GameServer를 할당 및 반환, 업데이트를 할 수 있음
- 기본 시나리오
 - k8s 클러스터가 dedicated VM 또는 베어메탈이라고 가정할 때
 - FPS게임서버(프로세스)이 2CPU 56코어(112스레드) 베어메탈에 프로비저닝 된다.
 - 방마다 접속자가 차면 새로운 프로세스가 생성되고 코어당 수십명씩 접속하여 플레이를 한다.
 - 게임 서버개발자는 심각한 버그를 파악하고 전체 서버에 패치를 배포하고자 한다.
 - 모든 서버에 배포를 하기엔 운영측면에서는 어렵다.
 - 서버 프로그램을 수정하고 릴리즈를 다시 하고 모든 서버에 사용자가 없음을 확인한 후에 배포를 한다.
- Agones를 활용한다면?
 - container로 게임프로세스를 물리 core당 하나씩 띄운다 (k8s + agones활용)
 - Agones를 활용하여 서버 중단없이 new version 배포 가능 (GameServerFleet)
 - Canary배포 컨셉을 게임서버에서도 구현. 거기에 **Spinnaker** 와 같은 CD도구까지 활용한다면!!!
 - 생성될때마다 매치메이킹 서버가 유저들을 하나의 서버프로세스로 udp로 접속시키고 모든 CPU 코어가 소모되면 다른 서버(노드)를 프로비저닝 (Node Scaling)

Spinnaker

- OpenSource multi-cloud CD platform
- Initially developed by Netflix's Asgard (2014)
 - Open-Sourced in 2015
- Designed with pluggability in mind
- Support for all major Cloud Provider
 - openstack, GCP, Azure, AWS, IKS, Kubernetes



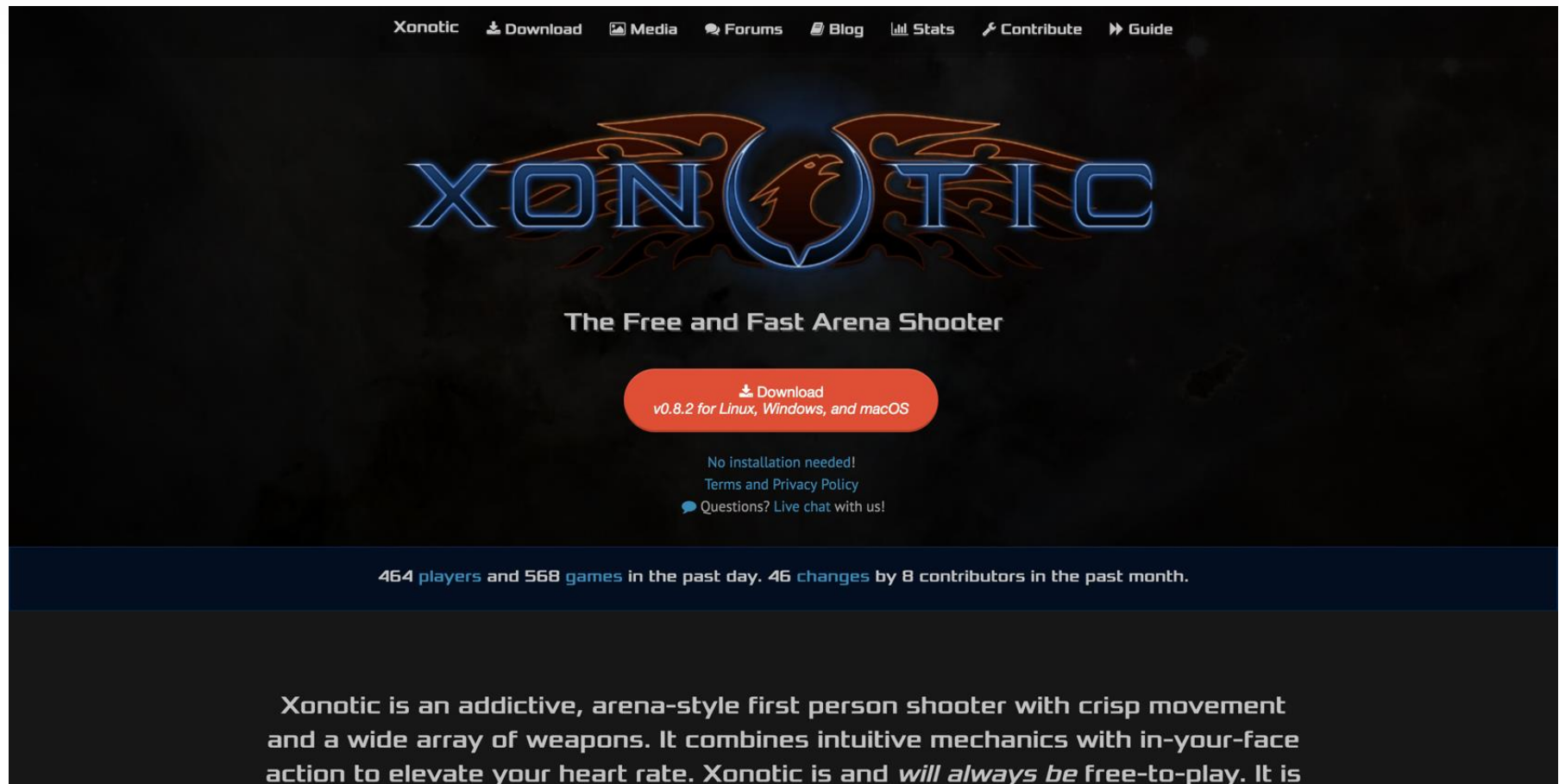
Spinnaker on Agones



- Agones - Spinnaker 활용방안 (Continuous Delivery)
 - 게임서버 배포 (Agones)
 - 게임서버 수정 및 배포 (git)
 - Git Webhook을 통한 Spinnaker Pipeline구성
 - Patch Manifests (Scale-out, 신규버전 배포)

Agones Demo

- Xonotic - OpenSource FPS Game
<http://www.xonotic.org/>



Conclusion

- 아직 초기단계, 하지만 실시간 멀티플레이 게임에서의 k8s도입은 가치가 있어보임
- 고성능 VM이나 베어메탈로 서비스중인 게임사는 R&D 진행해보는것도...
- Dedicated Server 개념이다 보니 Host Port를 사용하고 ingress나 loadbalancer 를 사용하지 않으므로 IP DDoS공격에 대한 대응책이 필요함
- Cloud Z Labs - <http://cloudz-labs.sk.com/>
운영 중인 애플리케이션을 Cloud Application으로 실제 전환하면서 Cloud Application 개발 역량을 습득하도록 도와주는 **"코칭 프로그램"**
 - Spring Boot, MSA, DevOps, Refactoring, PaaS, Kubernetes 등
- Cloud Z Labs - <http://cloudz-labs.sk.com/game>
 - 인디게임사 Challenge지원 및 게임 개발 협업프로그램 운영
 - Agones 및 Serverless 활용한 게임개발 연구
 - Out-Game MSA 구현



감사합니다

rokakjw@sk.com

zaction@sk.com

ddiwoong@gmail.com

Facebook @ddiwoong

