Google Cloud

Enterprise Grade Minecraft On Kubernetes

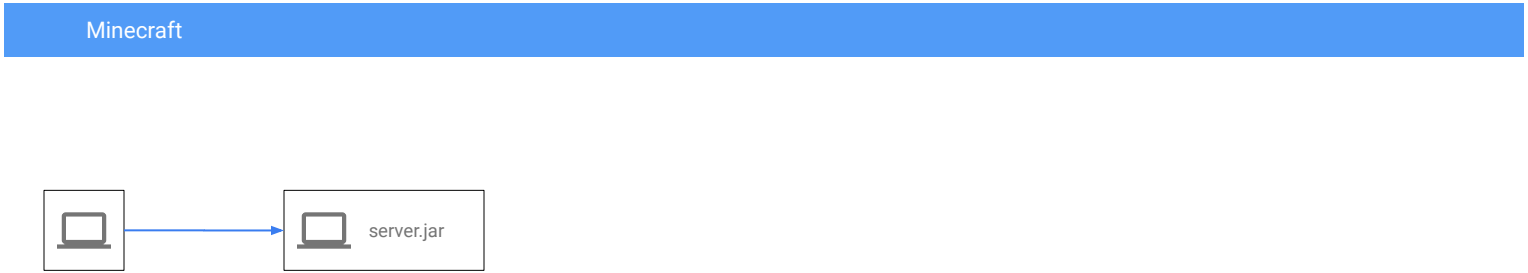
Friday, October 28

Because why not?

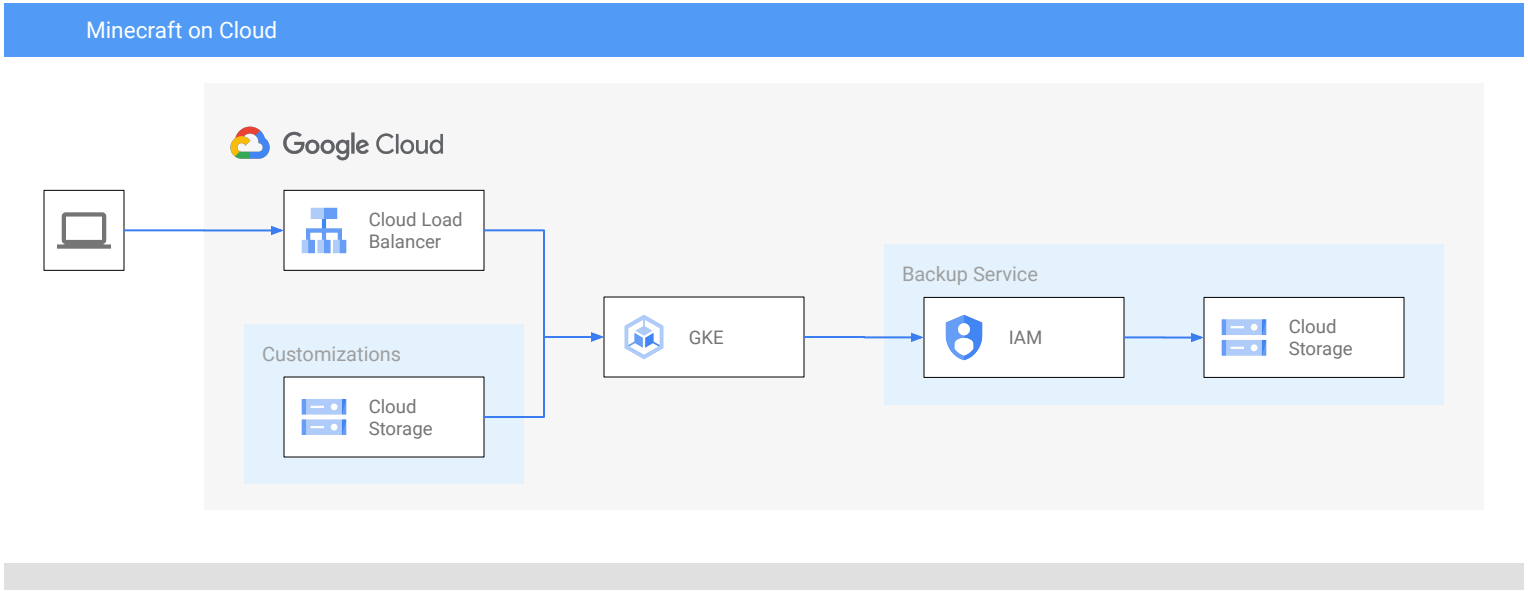
Casey West

Developer Advocate
Google Cloud Engineering
caseywest@google.com
@caseywest
he/him

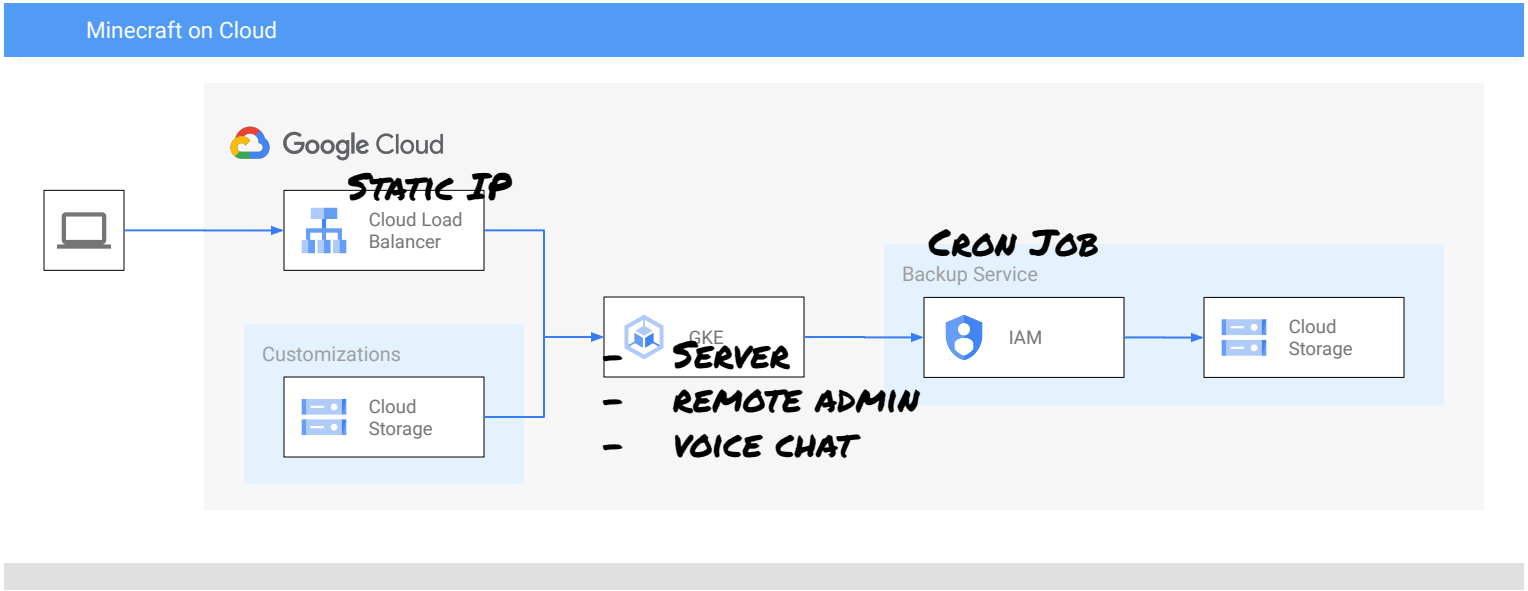
The Architecture Diagram



The Architecture Diagram on Cloud



The Architecture Diagram on Cloud (for real)



Create a Cluster

Specify our "fast-disk" storage class in a persistent volume claim to manage state for this service.

```
$ gcloud container clusters create minecraft-cluster
  --zone us-central1-a --release-channel stable
  --machine-type c2-standard-4 --num-nodes 1
  --enable-autoupgrade --enable-autorepair
```

State management

State: StorageClass

This service performs better when disks are faster. Solid State Disks (SSDs) are good.

```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: fast-disk
provisioner: kubernetes.io/gce-pd
volumeBindingMode: WaitForFirstConsumer
parameters:
  type: pd-ssd
reclaimPolicy: Retain
allowVolumeExpansion: true
```

State: PersistentVolumeClaim

ReadWriteOncePod ensures only a single pod can access the volume.

```yaml
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: minecraft-minecraft-datadir
  labels:
    app: minecraft-minecraft
  annotations:
    volume.beta.kubernetes.io/storage-class: "fast-disk"
spec:
  accessModes:
    - ReadWriteOncePod
  resources:
    requests:
      storage: "50Gi"
  storageClassName: "fast-disk"
```

Secret

This is the super secret password for RCON, a remote control protocol for game servers that can be enabled in this service.

```yaml
---
apiVersion: v1
kind: Secret
metadata:
  name: minecraft-minecraft
  labels:
    app: minecraft-minecraft
type: Opaque
data:
  rcon-password: "rotflmao"
```

Application deployment

Deployment

Disable rolling updates for this disk-based stateful service.

```yaml
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: minecraft-minecraft
  labels:
    app: minecraft-minecraft
spec:
  strategy:
    type: Recreate
  selector:
    matchLabels:
      app: minecraft-minecraft
  template:
    metadata:
      labels:
        app: minecraft-minecraft
    spec:
```

Deployment: containers, resources, volumes

Request as much CPU and memory as we can get.

Mount the fast disk for stateful storage.

```
spec:
  containers:
  - name: minecraft-minecraft
    image: "itzg/minecraft-server:latest"
    imagePullPolicy: Always
    resources:
      requests:
        cpu: 1000m
        memory: 5Gi
    volumeMounts:
    - name: datadir
      mountPath: /data
  volumes:
  - name: datadir
    persistentVolumeClaim:
      claimName: minecraft-minecraft-datadir
```

Deployment: env

Configure the service using environment variables, the way the vendor designed it.

```
spec:
  containers:
  - name: minecraft-minecraft
    env:
    - name: EULA
      value: "TRUE"
    - name: MEMORY
      value: "4G"
    - name: JVM_XX_OPTS
      value: "-XX:+UseG1GC -XX:+ParallelRefProcEnabled
-XX:MaxGCPauseMillis=200 -XX:+UnlockExperimentalVMOptions
-XX:+DisableExplicitGC -XX:-OmitStackTraceInFastThrow"
    - name: ENABLE_RCON
      value: "true"
    - name: RCON_PASSWORD
      valueFrom:
        secretKeyRef:
          name: minecraft-minecraft
          key: rcon-password
```

Deployment: ports

Ensure we expose all network ports from this application using its hard-coded ports and correct protocols.

```
spec:
  containers:
  - name: minecraft-minecraft
    ports:
    - name: minecraft
      containerPort: 25565
      protocol: TCP
    - name: rcon
      containerPort: 25575
      protocol: TCP
    - name: voicechat
      containerPort: 24454
      protocol: UDP
```

Deployment: probes

Let Kubernetes ensure the app is healthy.

```
spec:
  containers:
  - name: minecraft-minecraft
    readinessProbe:
      tcpSocket:
        port: 25565
      initialDelaySeconds: 30
      periodSeconds: 5
      failureThreshold: 10
      successThreshold: 1
      timeoutSeconds: 1
    livenessProbe:
      tcpSocket:
        port: 25565
      initialDelaySeconds: 30
      periodSeconds: 5
      failureThreshold: 10
      successThreshold: 1
      timeoutSeconds: 1
```

Deployment: initContainers

When deploying or updating the application, run customizations before the application is started.

When modifying state on disk, ensure the disk is mounted and writeable.

```
spec:
  initContainers:
    - args:
      - mkdir -p /data/mods; curl -o
/data/mods/voicechat.jar -L
https://storage.googleapis.com/BUCKET/voicechat-fabric-1.19
.2-2.3.12.jar;
      command:
      - /bin/sh
      - -c
      image: curlimages/curl
      name: install-voicechat
      volumeMounts:
      - mountPath: /data
        name: datadir
        readOnly: false
```

Service availability

Service: Static External IP Address

Consistent network addressability is handy.

```
$ gcloud compute addresses create mc-ip
  --region=us-central1-a

$ gcloud compute addresses describe mc-ip
```

Service: minecraft

Bind the primary network endpoint to the static IP.

```yaml
---
apiVersion: v1
kind: Service
metadata:
  name: minecraft-minecraft
  labels:
    app: minecraft-minecraft
spec:
  type: LoadBalancer
  loadBalancerIP: 34.XX.XXX.XXX
  externalTrafficPolicy: Cluster
  ports:
  - name: minecraft
    port: 25565
    targetPort: minecraft
    protocol: TCP
  selector:
    app: minecraft-minecraft
```

Service: rcon

Bind the remote administration protocol.

```yaml
---
apiVersion: v1
kind: Service
metadata:
  name: "minecraft-minecraft-rcon"
  labels:
    app: minecraft-minecraft
spec:
  type: LoadBalancer
  loadBalancerIP: 34.XX.XXX.XXX
  ports:
  - name: rcon
    port: 25575
    targetPort: rcon
    protocol: TCP
  selector:
    app: minecraft-minecraft
```

Service: voicechat

Bind the voice chat feature we added.

```yaml
---
apiVersion: v1
kind: Service
metadata:
 name: minecraft-minecraft-voicechat
 labels:
   app: minecraft-minecraft-voicechat
spec:
 type: LoadBalancer
 loadBalancerIP: 34.XX.XXX.XXX
 externalTrafficPolicy: Cluster
 ports:
 - name: voicechat
   port: 24454
   targetPort: voicechat
   protocol: UDP
 selector:
   app: minecraft-minecraft
```

Data backup

Backups: ServiceAccount

Create a Kubernetes service account.

```
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: minecraft-backup-runner-sa
  annotations:
    iam.gke.io/gcp-service-account:
minecraft-backup-cronjob@PROJECT.iam.gserviceaccount.com
```

Backups: Role

Create a role in Kubernetes that can interact with PODs.

```
---
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: minecraft-backup-runner
rules:
- apiGroups:
  - "*"
  resources:
  - pods
  - pods/exec
  verbs:
  - 'list'
  - 'get'
  - 'create'
```

Backups: RoleBinding

Bind your Kubernetes role to your Kubernetes service account for POD management.

```yaml
---
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: minecraft-backup-runner
subjects:
- kind: ServiceAccount
  name: minecraft-backup-runner-sa
roleRef:
  kind: Role
  name: minecraft-backup-runner
  apiGroup: ""
```

Backups: IAMServiceAccount

Create a Service Account for your infrastructure provider.

```
---
apiVersion: iam.cnrm.cloud.google.com/v1beta1
kind: IAMServiceAccount
metadata:
  name: minecraft-backup-cronjob
spec:
  displayName: Service account for Minecraft Server Backups
```

Backups: IAMPolicyMember

Add the IAM service account to a policy that can manage Storage Objects.

```
---
apiVersion: iam.cnrm.cloud.google.com/v1beta1
kind: IAMPolicyMember
metadata:
  name: policy-member-binding
spec:
  member:
serviceAccount:minecraft-backup-cronjob@PROJECT.iam.gservic
eaccount.com
  role: roles/storage.objectAdmin
  resourceRef:
    apiVersion:
resourcemanager.cnrm.cloud.google.com/v1beta1
    kind: Project
    external: projects/PROJECT
```

Backups: IAMPolicy

Create a policy that allows a Kubernetes workload (POD) identify as an infrastructure provider IAM Service Account.

```
---
apiVersion: iam.cnrm.cloud.google.com/v1beta1
kind: IAMPolicy
metadata:
  name: iampolicy-workload-identity-minecraft-backup-cronjob
spec:
  resourceRef:
    apiVersion: iam.cnrm.cloud.google.com/v1beta1
    kind: IAMServiceAccount
    name: minecraft-backup-cronjob
  bindings:
    - role: roles/iam.workloadIdentityUser
      members:
        -
serviceAccount:PROJECT.svc.id.goog[default/minecraft-backup-runne
r-sa]
```

Backups: CronJob

Define a regular backup twice a day.

Ensure we don't try to backup synchronously.

```
---
apiVersion: batch/v1
kind: CronJob
metadata:
  name: minecraft-backup
spec:
  schedule: "0 */12 * * *"
  concurrencyPolicy: Forbid
  jobTemplate:
    spec:
      template:
```

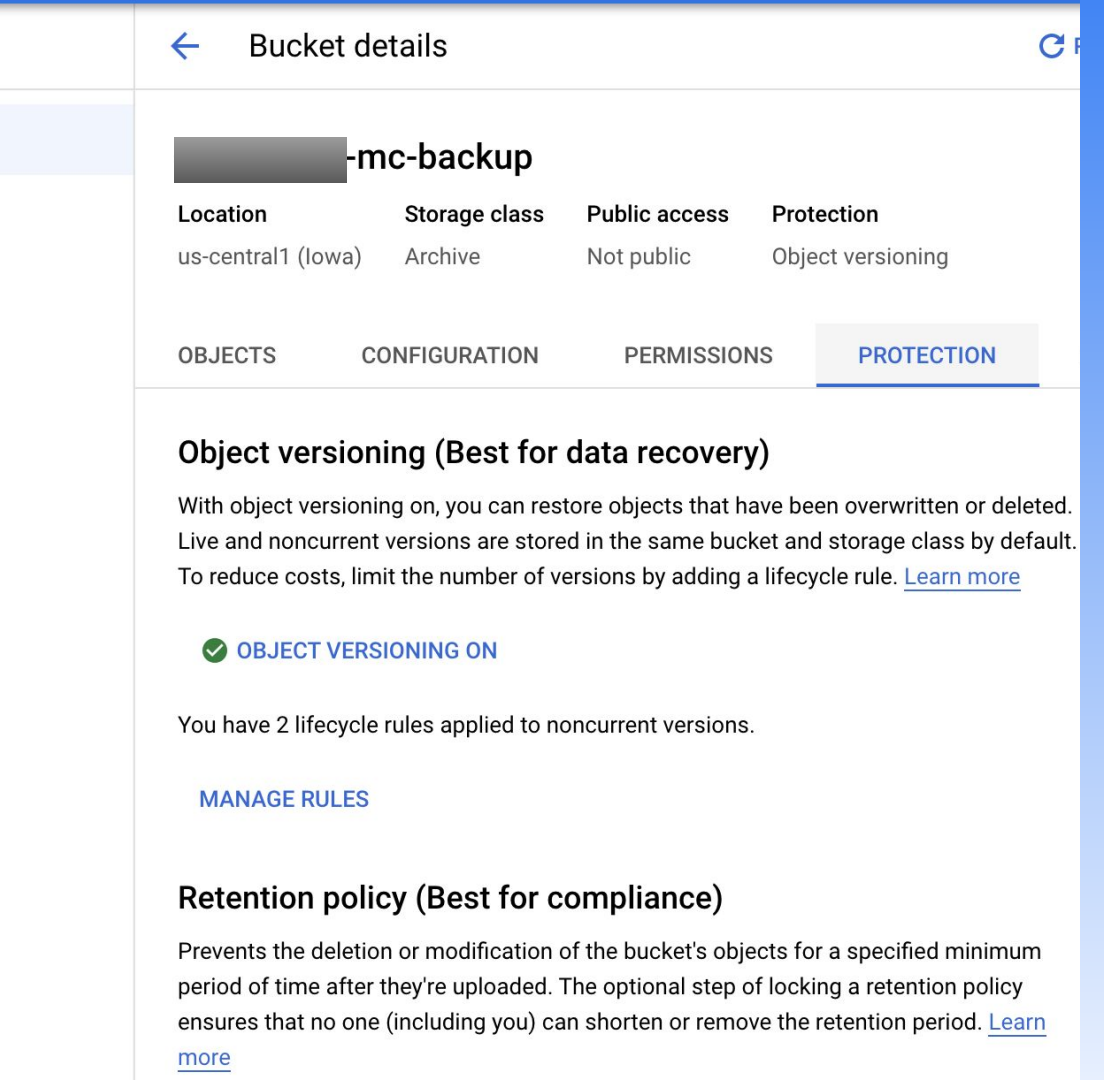
Backups: template.spec

Safely archive on-disk data.

Move it to long-term storage.

If the job fails, try again.

```
spec:
  serviceAccountName: minecraft-backup-runner-sa
  containers:
  - name: cloud-sdk
    image: google/cloud-sdk:alpine
    command: [ "/bin/sh", "-c" ]
    args:
    -
      gcloud components install kubectl gsutil;
      POD_ID=$(kubectl get pods
--selector=app=minecraft-minecraft
--field-selector=status.phase=Running -o
jsonpath={.items..metadata.name});
      kubectl exec ${POD_ID} -- rcon-cli save-off;
      kubectl exec ${POD_ID} -- rcon-cli save-all;
      kubectl exec ${POD_ID} -- tar cf - /data | gzip > data.tgz;
      kubectl exec ${POD_ID} -- rcon-cli save-on;
      gsutil cp data.tgz gs://BUCKET
  restartPolicy: OnFailure
```



Backup Storage

Images help do the heavy-lifting in a slide. This allows us to rely less on copy, or to highlight the image or graphic. Images can be used to illustrate, educate, or even entertain in ways that copy cannot. So give some thought to the image you choose.

Again, keep the copy to approximately this length. Be mindful that the copy aligns with the image, and make your point as concisely as possible.

App runs?

Service Connects

Service is Available

Use the Helm Charts

https://github.com/itzg/minecraft-server-charts

Thank you.

Google Cloud