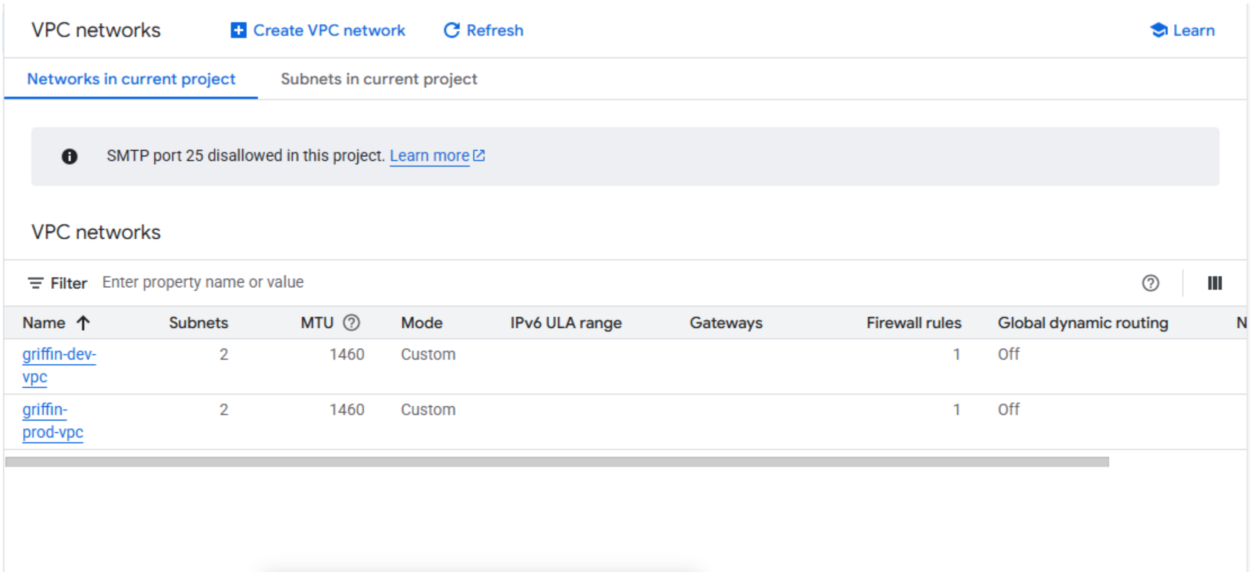
Develop your Google Cloud Network: Challenge Lab

Task 1. Create development VPC manually

- Create a VPC called griffin-dev-vpc with the following subnets only:
  - griffin-dev-wp
    - IP address block: 192.168.16.0/20
  - griffin-dev-mgmt
    - IP address block: 192.168.32.0/20


VPC networks
Create VPC network
Refresh
Learn
Networks in current project
Subnets in current project
Filter
Enter property name or value

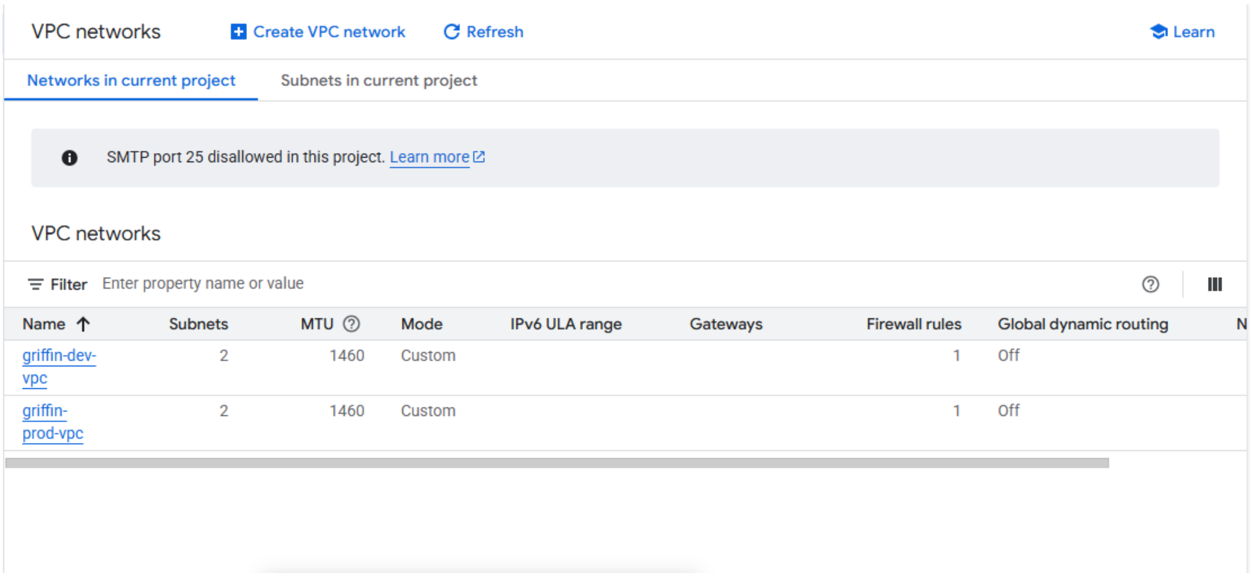
SMTP port 25 disallowed in this project. Learn more

VPC networks

Name	Subnets	MTU	Mode	IPv6 ULA range	Gateways	Firewall rules	Global dynamic routing	N
griffin-dev-vpc	2	1460	Custom			1	Off	
griffin-prod-vpc	2	1460	Custom			1	Off	

Task 2. Create production VPC manually

- Create a VPC called griffin-prod-vpc with the following subnets only:
  - griffin-prod-wp
    - IP address block: 192.168.48.0/20
  - griffin-prod-mgmt
    - IP address block: 192.168.64.0/20


VPC networks | Create VPC network | Refresh | Learn

Networks in current project | Subnets in current project

SMTP port 25 disallowed in this project. Learn more

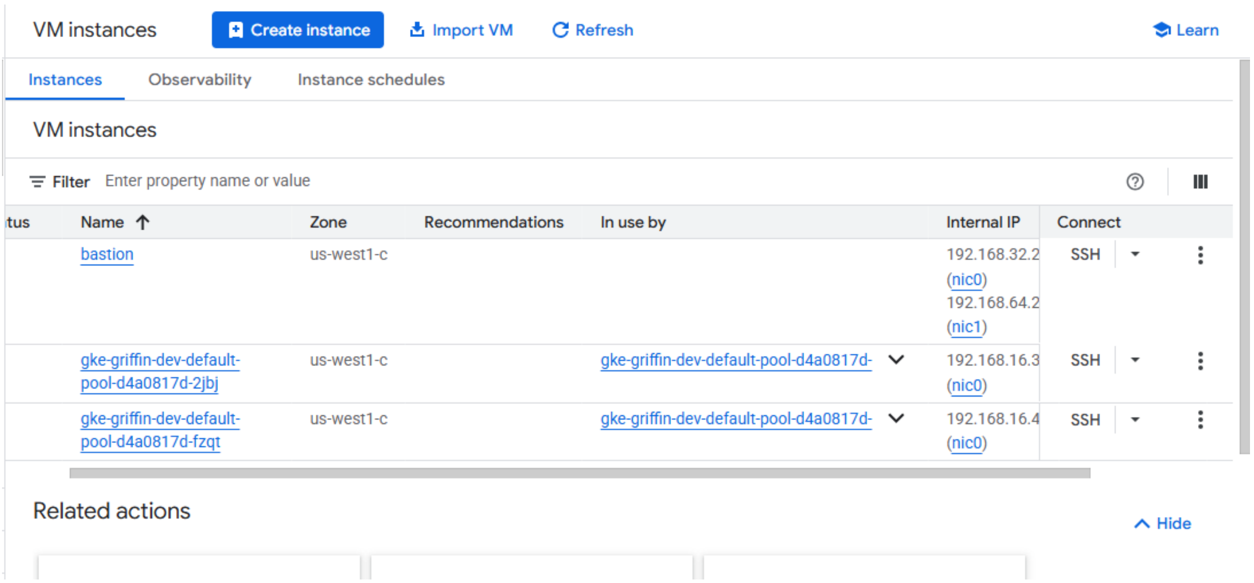
VPC networks

Filter Enter property name or value

Name	Subnets	MTU	Mode	IPv6 ULA range	Gateways	Firewall rules	Global dynamic routing	N
griffin-dev-vpc	2	1460	Custom			1	Off	
griffin-prod-vpc	2	1460	Custom			1	Off	

Task 3. Create bastion host

- Create a bastion host with two network interfaces, one connected to griffin-dev-mgmt and the other connected to griffin-prod-mgmt. Make sure you can SSH to the host.


VM instances | Create instance | Import VM | Refresh | Learn

Instances | Observability | Instance schedules

VM instances

Filter Enter property name or value

tus	Name	Zone	Recommendations	In use by	Internal IP	Connect
	bastion	us-west1-c			192.168.32.2 (nic0) 192.168.64.2 (nic1)	SSH
	gke-griffin-dev-default-pool-d4a0817d-2jbj	us-west1-c		gke-griffin-dev-default-pool-d4a0817d-	192.168.16.3 (nic0)	SSH
	gke-griffin-dev-default-pool-d4a0817d-fzqt	us-west1-c		gke-griffin-dev-default-pool-d4a0817d-	192.168.16.4 (nic0)	SSH

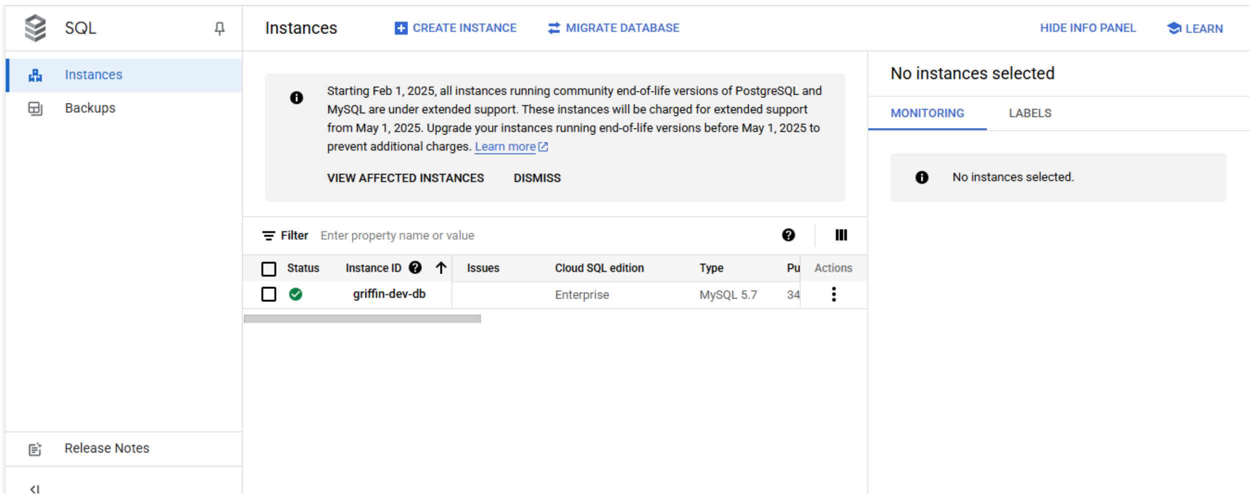
Related actions | Hide

Task 4. Create and configure Cloud SQL Instance

1. Create a **MySQL Cloud SQL Instance** called griffin-dev-db in REGION.
2. Connect to the instance and run the following SQL commands to prepare the **WordPress** environment:

CREATE DATABASE wordpress;
CREATE USER "wp_user"@"%" IDENTIFIED BY "stormwind_rules";
GRANT ALL PRIVILEGES ON wordpress.* TO "wp_user"@"%";
FLUSH PRIVILEGES;

These SQL statements create the worpdress database and create a user with access to the wordpress database.



Starting Feb 1, 2025, all instances running community end-of-life versions of PostgreSQL and MySQL are under extended support. These instances will be charged for extended support from May 1, 2025. Upgrade your instances running end-of-life versions before May 1, 2025 to prevent additional charges. Learn more

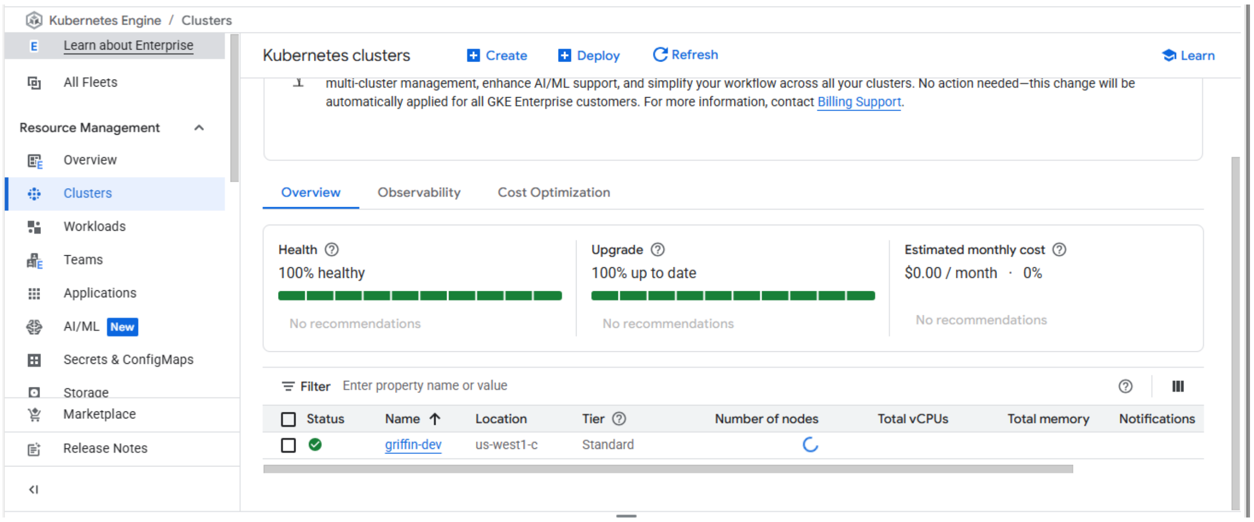
No instances selected

No instances selected.

Status	Instance ID	Issues	Cloud SQL edition	Type	Pu	Actions
✓	griffin-dev-db		Enterprise	MySQL 5.7	34	⋮

Task 5. Create Kubernetes cluster

- Create a 2 node cluster (e2-standard-4) called griffin-dev, in the griffin-dev-wp subnet, and in zone ZONE.



multi-cluster management, enhance AI/ML support, and simplify your workflow across all your clusters. No action needed—this change will be automatically applied for all GKE Enterprise customers. For more information, contact Billing Support.

Health: 100% healthy — No recommendations
Upgrade: 100% up to date — No recommendations
Estimated monthly cost: $0.00 / month · 0% — No recommendations

Status	Name	Location	Tier	Number of nodes	Total vCPUs	Total memory	Notifications
✓	griffin-dev	us-west1-c	Standard				

Task 6. Prepare the Kubernetes cluster

1. From Cloud Shell copy all files from gs://cloud-training/gsp321/wp-k8s.

The **WordPress** server needs to access the MySQL database using the *username* and *password* you created in task 4.

2. You do this by setting the values as secrets. **WordPress** also needs to store its working files outside the container, so you need to create a volume.
3. Add the following secrets and volume to the cluster using wp-env.yaml.
4. Make sure you configure the *username* to wp_user and *password* to stormwind_rules before creating the configuration.

You also need to provide a key for a service account that was already set up. This service account provides access to the database for a sidecar container.

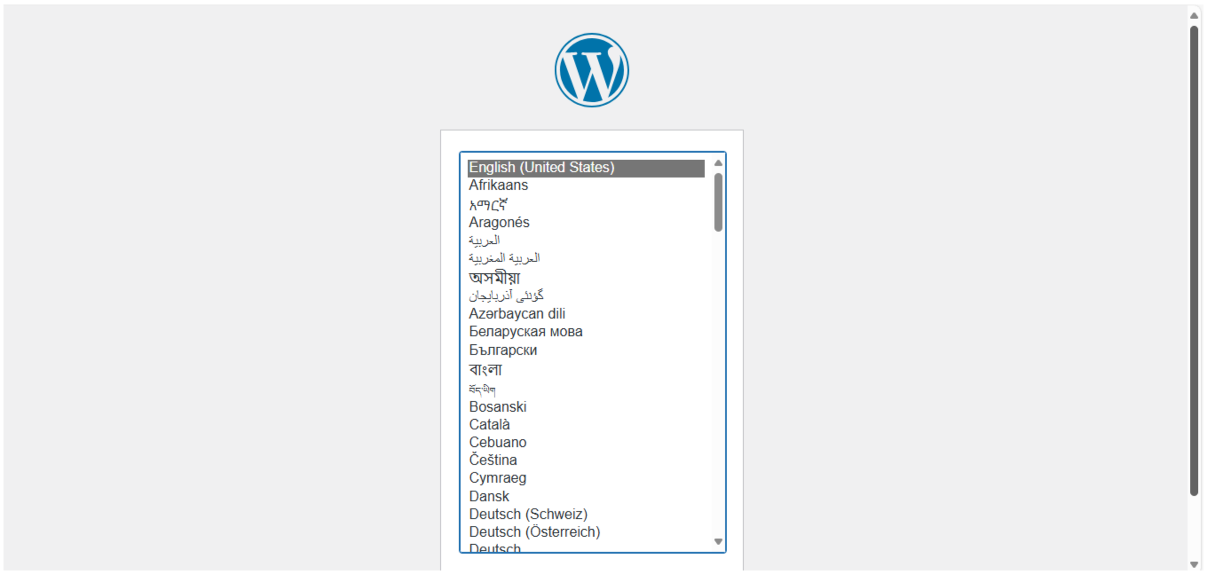
5. Use the command below to create the key, and then add the key to the Kubernetes environment:

```
gcloud iam service-accounts keys create key.json \
  --iam-account=cloud-sql-proxy@$GOOGLE_CLOUD_PROJECT.iam.gserviceaccount.com
kubectl create secret generic cloudsql-instance-credentials \
  --from-file key.json
```

Task 7. Create a WordPress deployment

Now that you have provisioned the MySQL database, and set up the secrets and volume, you can create the deployment using wp-deployment.yaml.

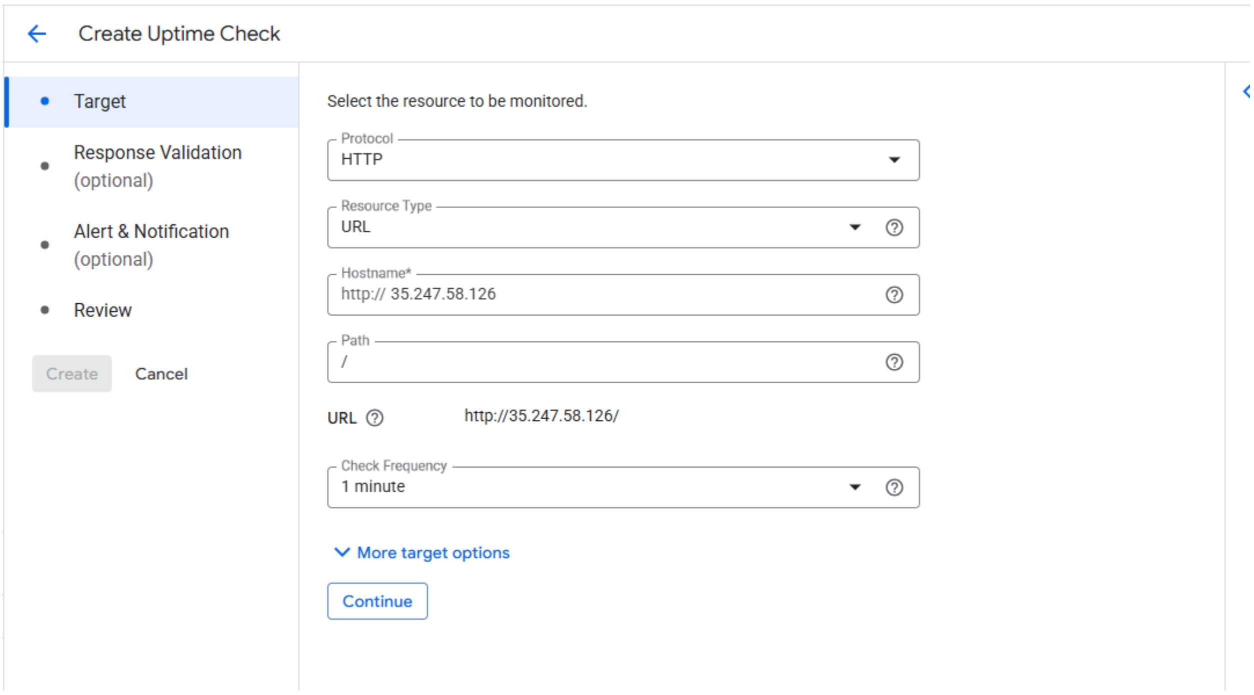
1. Before you create the deployment you need to edit wp-deployment.yaml.
2. Replace **YOUR_SQL_INSTANCE** with griffin-dev-db's **Instance connection name**.
3. Get the **Instance connection name** from your Cloud SQL instance.
4. After you create your WordPress deployment, create the service with wp-service.yaml.
5. Once the Load Balancer is created, you can visit the site and ensure you see the **WordPress** site installer.
   At this point the dev team will take over and complete the install and you move on to the next task.



English (United States)
Afrikaans
አማርኛ
Aragonés
العربية
العربية المغربية
অসমীয়া
گؤنئی آذربایجان
Azərbaycan dili
Беларуская мова
Български
বাংলা
བོད་ཡིག
Bosanski
Català
Cebuano
Čeština
Cymraeg
Dansk
Deutsch (Schweiz)
Deutsch (Österreich)
Deutsch

Task 8. Enable monitoring

- Create an uptime check for your WordPress development site.



Create Uptime Check
Target
Response Validation (optional)
Alert & Notification (optional)
Review
Create
Cancel

Select the resource to be monitored.

Protocol: HTTP

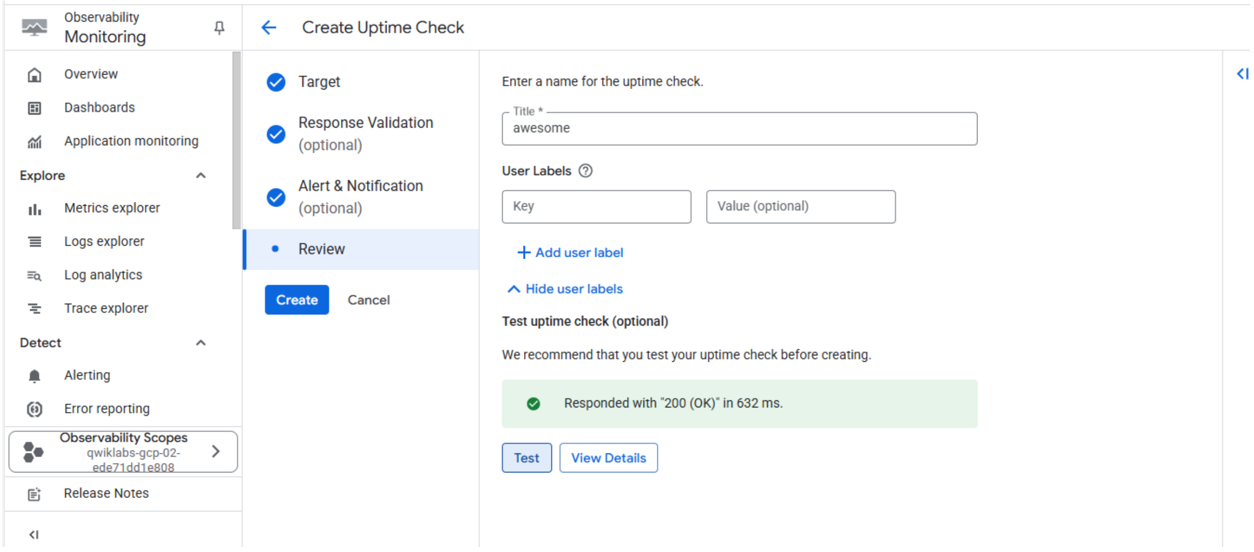
Resource Type: URL

Hostname*: http:// 35.247.58.126

Path: /

URL: http://35.247.58.126/

Check Frequency: 1 minute



Enter a name for the uptime check.

Title * awesome

User Labels

Key | Value (optional)

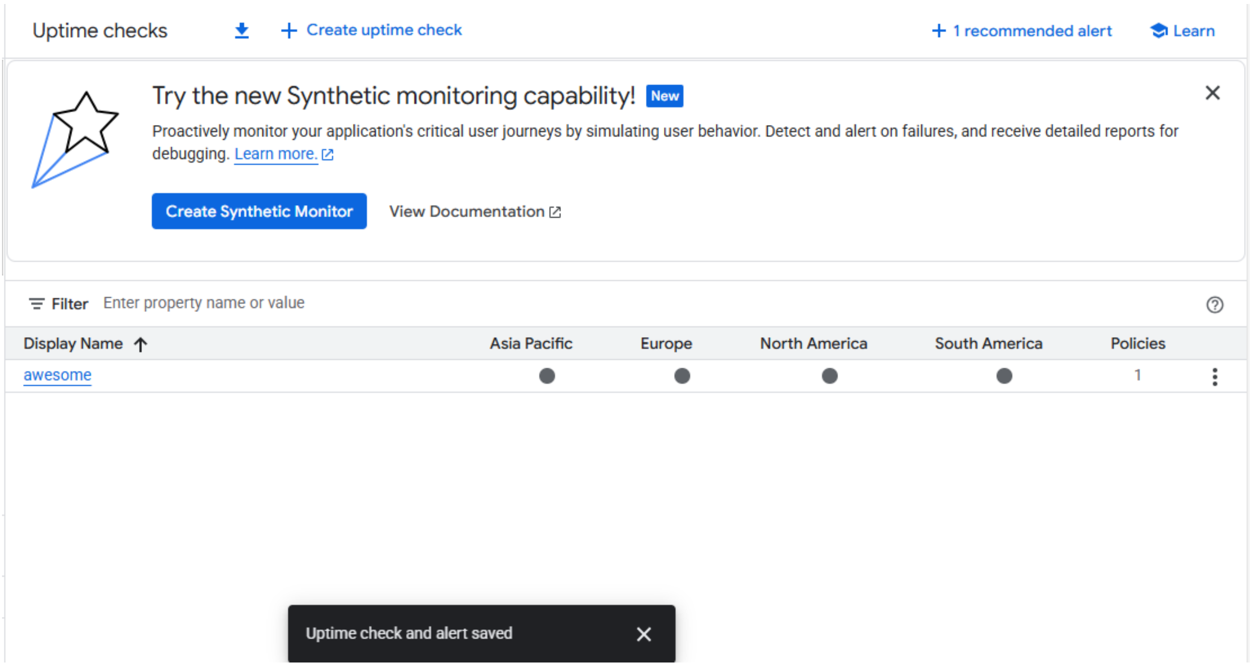
+ Add user label

Hide user labels

Test uptime check (optional)

We recommend that you test your uptime check before creating.

Responded with "200 (OK)" in 632 ms.



Proactively monitor your application's critical user journeys by simulating user behavior. Detect and alert on failures, and receive detailed reports for debugging. Learn more.

Display Name	Asia Pacific	Europe	North America	South America	Policies
awesome	●	●	●	●	1

Uptime check and alert saved

Task 9. Provide access for an additional engineer

- You have an additional engineer starting and you want to ensure they have access to the project. Grant them the editor role to the project.

The second user account for the lab represents the additional engineer.