Boon AI Deployment Documentation

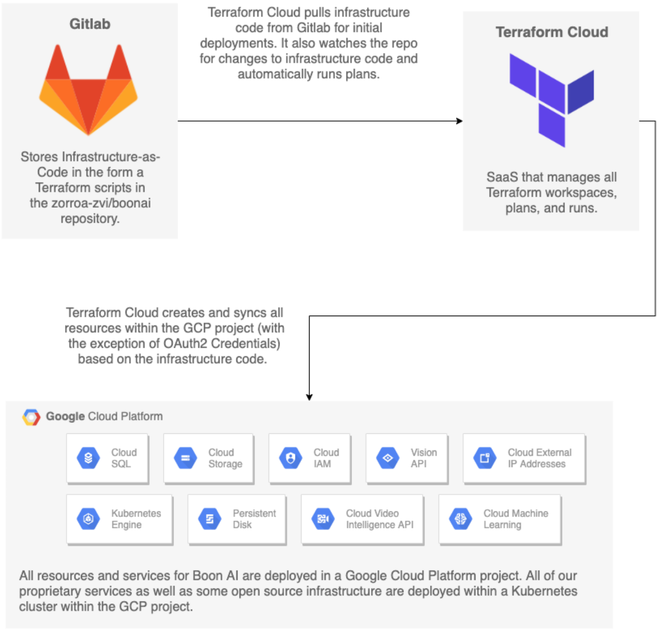
Overview

Introduction

Boon AI is complex SaaS consisting of multiple internal services, console UI and a publicly consumable REST API. Currently deployment is only supported on the Google Cloud Platform (GCP). The infrastructure consists of several native GCP resources including SQL, Cloud Storage and Google Kubernetes Engine (GKE). All of our proprietary services are deployed within a Kubernetes cluster running on GKE.

We utilize Infrastructure-as-Code (IaC) as much as possible in the deployment process. Terraform is the main tool utilized for IaC processes. It's possible to run Terraform from a local machine but we recommend using their SaaS service, Terraform Cloud, for managing and running Terraform in order to keep the deployments centralized, shareable and secure.

Deployment Architecture


Gitlab
Stores Infrastructure-as-Code in the form a Terraform scripts in the zorroa-zvi/boonai repository.
Terraform Cloud pulls infrastructure code from Gitlab for initial deployments. It also watches the repo for changes to infrastructure code and automatically runs plans.
Terraform Cloud
SaaS that manages all Terraform workspaces, plans, and runs.
Terraform Cloud creates and syncs all resources within the GCP project (with the exception of OAuth2 Credentials) based on the infrastructure code.
Google Cloud Platform
Cloud SQL
Cloud Storage
Cloud IAM
Vision API
Cloud External IP Addresses
Kubernetes Engine
Persistent Disk
Cloud Video Intelligence API
Cloud Machine Learning
All resources and services for Boon AI are deployed in a Google Cloud Platform project. All of our proprietary services as well as some open source infrastructure are deployed within a Kubernetes cluster within the GCP project.

*NOTE: This does not represent the SDLC of the code. This diagram only outlines the mechanisms used to deploy infrastructure.

Step-By-Step Deployment Guide

This guide will walk you through the entire process of standing up a new deployment of Boon AI.

→ Secure a Domain /1.-secure-a-domain

→ Prepare the GCP Project /2.-prepare-the-gcp-project

→ Configure Terraform Workspace /3.-configure-terraform-workspace

→ Execute Terraform Plan /4.-execute-terraform-plan

→ Configure DNS /5.-configure-dns

→ Change Default Admin Password /6.-change-default-admin-password

GCP Marketplace Integration

Boon AI has iSaaS integration with the Google Marketplace. This means that users can purchase access to the production Boon AI instance via the Google Marketplace. When a user subscribes via the Marketplace an account is provisioned for them and a new project is created. The link below explains how that integration is configured and deployed.

→ Overview /

Step-By-Step Deployment Guide

Secure a Domain

Boon AI requires two external URLs. You'll need to secure a domain name for these urls. It is recommended that you get one domain name for the deployment and use two sub-domains for the urls. The two urls required are discussed below.

Boon AI Console URL

This url will direct users to the Boon AI Console which is a web application for managing the Boon AI platform. An example would be https://boonai.app which is used for our SaaS production deployment.

Boon AI REST API URL

This url hosts the REST API that allows users to integrate with Boon AI. An example would be https://api.boonai.app which is used for our SaaS production deployment.

> ⓘ Once you have decided on these two urls make a note of them. You will need them later in the process.

Information Required For The Next Step

- ☐ Boon AI Console Fully Qualified Domain Name
- ☐ Boon AI API Fully Qualified Domain Name

Prepare the GCP Project

Prerequisites

- Access to the GCP console at https://console.cloud.google.com/.
- User permissions that allow creating new projects.

Create a project

Use the guide below to create a new project. Give the project a name that is descriptive.

Creating and managing projects | Resource Manager Documentation https://cloud.google.com/resource-manager/docs/creating-managing-projects#console

> Once the project is created note the Project ID (which may be different from the Project Name) and save it for later. You will need it later on in the process.

Create a service account for Terraform

Terraform needs credentials in order to deploy all of the required resources. We'll be creating a service account for Terraform to use.

Use the guide below to create a new service account. When creating the account input the following information.

Creating and managing service accounts | Cloud IAM Documentation https://cloud.google.com/iam/docs/creating-managing-service-accounts#iam-service-accounts-create-console

- Service account name: `terraform`
- Service account description: `Used by Terraform Cloud for infrastructure deployment.`
- Role: `Owner` - You can find this in the `Project` section when browsing roles.

> Save the email address associated with this service account. You will need it later in the process.

Create a service account key

In order for Terraform to use the service account we just created we need to issue a key for it to pass along for authentication.

Use the guide below to create a service account key. When prompted choose to create a `JSON` account key.

Creating and managing service account keys | Cloud IAM Documentation https://cloud.google.com/iam/docs/creating-managing-service-account-keys#iam-service-account-keys-create-console

ⓘ The JSON service account key should have downloaded to your machine. Note the location of this file, you will need to reference the contents of it later.

Configure the OAuth Consent Screen

The Boon AI console allows users with Google accounts to provision accounts and log in via OAuth 2.0. When a user logs in with this method the first time a consent screen is displayed. We need to handle the configuration of that screen in order for it to work.

Go to the GCP Console tool below to configure the OAuth Consent Screen. Use the following information as a guide for the configuration. Based on your deployment these values may vary.

https://console.cloud.google.com/apis/credentials/consent https://console.cloud.google.com/apis/credentials/consent

- User/Application Type: Review the documentation in the tool and determine whether `Internal` or `External` is correct for your deployment.
- Application Name: `Boon AI`
- Application Logo: Download the Boon AI Logo from the link below.

Boon AI Logo BoonAI-mark-clear-square.png - 20KB

- Leave Support Email and Scopes alone.
- Authorized Domains: Top level domain you will be using for the Boon AI Console. For example the top level domain of `console.boonai.app` is `boonai.app` .
- Application Homepage Link: Fully qualified domain name (FQDN) to be used for the Boon AI Console.
- Application Privacy Policy Link: `<FQDN>/policies/20200414/privacy-policy.pdf` - FQDN is the same entry you used for Application Homepage Link.
- Leave Application Terms of Service Link blank.

Create OAuth 2.0 Client ID Credential

Now that we have the consent screen configured we need to set up a client ID the Boon AI Console will use when authenticating users via OAuth 2.

Use the guide below to create a Client ID. When creating the Client use the following information.

Setting up OAuth 2.0 - Google Cloud Platform Console Help https://support.google.com/cloud/answer/6158849?hl=en

- Application Type: `Web Application`
- Name: `Boon AI`
- Authorized JavaScript Origins: FQDN of the Boon AI Console.
- Authorized redirect URIs: FQDN of the Boon AI Console.

ⓘ After you've created the Client ID Credential save the actual Client ID, you will need it later in process. You only need to save the portion that comes before `apps.googleusercontent.com.`

Create Monitoring Workspace

In order for terraform to deploy custom monitoring dashboards the workspace needs to be set up. Simply go to the link below for the project and the workspace will start setting itself up.

https://console.cloud.google.com/monitoring/dashboards https://console.cloud.google.com/monitoring/dashboards

Optional: Request Higher Quotas for ML APIs

The default quotas for the ML apis for projects start out very low. Depending on which APIs you plan on using you may want to go ahead and request higher usage quotas now. If not you will likely see failed processing jobs due to quotas being hit.

Use the guide below to request increased quotas for GCP ML APIs.

Resource quotas | Compute Engine Documentation | Google Cloud https://cloud.google.com/compute/quotas

Information Required For The Next Step

- □ GCP Project ID
- □ Service Account email address
- □ Service Account Json Key file
- □ OAuth2 Client ID

Prepare Clarifai Application

Create Clarifai account and application.

Follow this quick start guide but stop once you've completed creating your first application. Name the application something indicative of the environment you are setting up (i.e. boon-ai-qa).

https://docs.clarifai.com/getting-started/quick-start https://docs.clarifai.com/getting-started/quick-start

---

Get an API key for your application.

Use this guide to create an API key for your new application.

App-Specific API Keys https://docs.clarifai.com/getting-started/authentication/app-specific-api-keys#create-an-api-key-in-portal

> ⓘ Hang on to your API key. It will be needed later in the process.

Prepare Azure Account

Sign up for an Azure account.

You can sign up for your account here.

Once your account is activated you'll need to upgrade the account to a paid plan and enter a valid credit card. There should be a large button at the top of the Azure portal prompting you to upgrade.

Create a Service Principal in the Azure Portal

Follow the guide below to create a Service Principal. This is what Terraform will use to create the required Azure resources.

> When asked to pick a role during the tutorial below choose "Contributor".

https://www.terraform.io/docs/providers/azurerm/guides/service_principal_client_secret.html#creating-a-service-principal-in-the-azure-portal

https://www.terraform.ic
_principal_client_secret.h
azure-portal

> After completing this tutorial be sure to keep the client_id, tenant_id, client_secret, and subscription_id handy for the next step.

Prepare AWS Account

Sign up for an AWS account.

Create a new account here.

Use the following guide to set up a payment method for your account.

🔗 **Managing your AWS payment methods - AWS Billing and Cost Management** https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/manage-payment-method.html#manage-view-credit

Create an admin user.

Use the guide below to create an administrator user that Terraform can use to provision the necessary resources.

🔗 **Creating your first IAM admin user and user group - AWS Identity and Access Management** https://docs.aws.amazon.com/IAM/latest/UserGuide/getting-started_create-admin-group.html

Create an access key for the admin user.

Go to the guide below and follow the instructions in the "To create, modify, or delete another IAM user's access keys (console)" section to create an access key for the admin user.

> ⓘ When you create the access key make sure to keep track of the access key id and secret key. You will need them later on and you cannot retrieve the secret key again.

🔗 **Managing access keys for IAM users - AWS Identity and Access Management** https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html#Using_CreateAccessKey

Configure Terraform Workspace

Prerequisites

- Terraform Cloud account.
- Permissions to create a new workspace.
- Password for the software@zorroa.com docker hub account. Can be found in 1Password.
- Password for SMTP email server.

Log In to the Zorroa Terraform Cloud Organization

Go to the link below and log in.

🔗 **https://app.terraform.io/app/zorroa/workspaces** https://app.terraform.io/app/zorroa/workspaces

---

Create a Workspace

Use the guide below to create a new workspace. Use the following information when creating the workspace.

🔗 **Terraform** https://www.terraform.io/docs/cloud/workspaces/creating.html

- Connect to a version control provider: Choose gitlab.com, it will be the only option.
- Choose a repository: zorroa-zvi/boonai
- Workspace name: Use the GCP Project ID as the workspace name.
- Advanced Options -> Terraform Working Directory: `deployment/terraform`
- Advanced Options -> VCS Branch: `master`

Configure Workspace Variables

Before we can execute a Terraform plan we need to configure a set of variables that will customize the plan for your deployment.

Use the guide below along with the table to set the variables in the workspace. In the table if the `Sensitive` column is checked it is VERY important to select the sensitive checkbox when creating the variable.

🔗 **Terraform** https://www.terraform.io/docs/cloud/workspaces/variables.html#managing-variables-in-the-ui

Variable Name	Value	Sensitive
project	GCP project ID.	
project-number	GCP project number.	

docker-username	Docker hub username the GKE cluster will use to pull images.	
docker-email	Docker hub user email the GKE cluster will use to pull images.	
docker-password	Password for the docker hub user.	☑
smtp-host	Hostname of the SMTP server.	
smtp-user	Username for the SMTP server.	
smtp-from-email	Default email address the SMTP server will use to send mail.	
smtp-password	Password for the smtp email server.	☑
terraform-credentials	Copy/paste the complete contents of the Service Account Json Key file.	☑
google-oauth-client-id	Google OAuth Client ID configured in Step 2. Do not include `.apps.googleusercontent.com` in the id.	
environment	This is by our error handling systems to segment errors by their environment. Choose something descriptive or use the GCP Project ID.	
wallet-domains	List of domain names for the Boon AI Console. Do not include `https://`	
zmlp-domains	List of domain names for the Boon AI REST API. Do no include `https://`	
clarifai-key	API key for Clarifai that has access to all scopes.	☑
aws-key	ID for an AWS access key associated with an account that has permissions for all ML services.	
aws-secret	Secret key for an AWS access key associated with an account that has permissions for all ML services.	☑
azure-subscription-id	Azure subscription ID to be used with Azure provider.	
azure-client-id	Client ID from the Azure service principal to be used with the Azure provider.	
azure-tenant-id	Tenant/directory ID of the Azure service principal to be used with the Azure provider.	
azure-client-secret	Secret key generated for the Azure service principal to be used with the Azure provider.	☑

Execute Terraform Plan

Queue a Plan

The first step in executing a Terraform run is to queue a plan. This will cause Terraform to introspect the GCP project and determine which resource need to be created (all of them in our case).

Use the guide below to manually queue a plan in your workspace. Be sure to fill out the reason for queueing the plan so others can understand why it happened. For this first plan you could write something like "Deploying infrastructure for the first time".

Terraform https://www.terraform.io/docs/cloud/run/ui.html#manually-starting-runs

---

Confirm and Run the Plan

Once the plan completes you will see a breakdown of all the resources it plans to create. It should be ~80. In order to actually start creating the resources you need to confirm and apply the plan.

Confirm and apply your plan using the guide below.

Terraform https://www.terraform.io/docs/cloud/run/ui.html#confirming-or-discarding-plans

---

Rinse and Repeat

We fall into the category of Terraform power-users and have a very complex plan with many dependencies and long running tasks. This results in many failures during the initial deploy. This is a known limitation of our setup and Terraform in general. Luckily the entire process is idempotent so there is no harm in rerunning it. You will likely need to execute the process above 3-5 times before getting a successful run. If you have run the plan in excess of 5 times without a full success please contact devops.

> ⓘ One of the longest running tasks is the kubernetes building and modifying itself. If a run has failed you can check on the cluster in the GCP Console. If the Boon AI cluster has a spinning icon it means it is updating. It's best to wait for the update to complete before attempting another Terraform run. The clusters can viewed at this link - https://console.cloud.google.com/kubernetes/list

Configure DNS

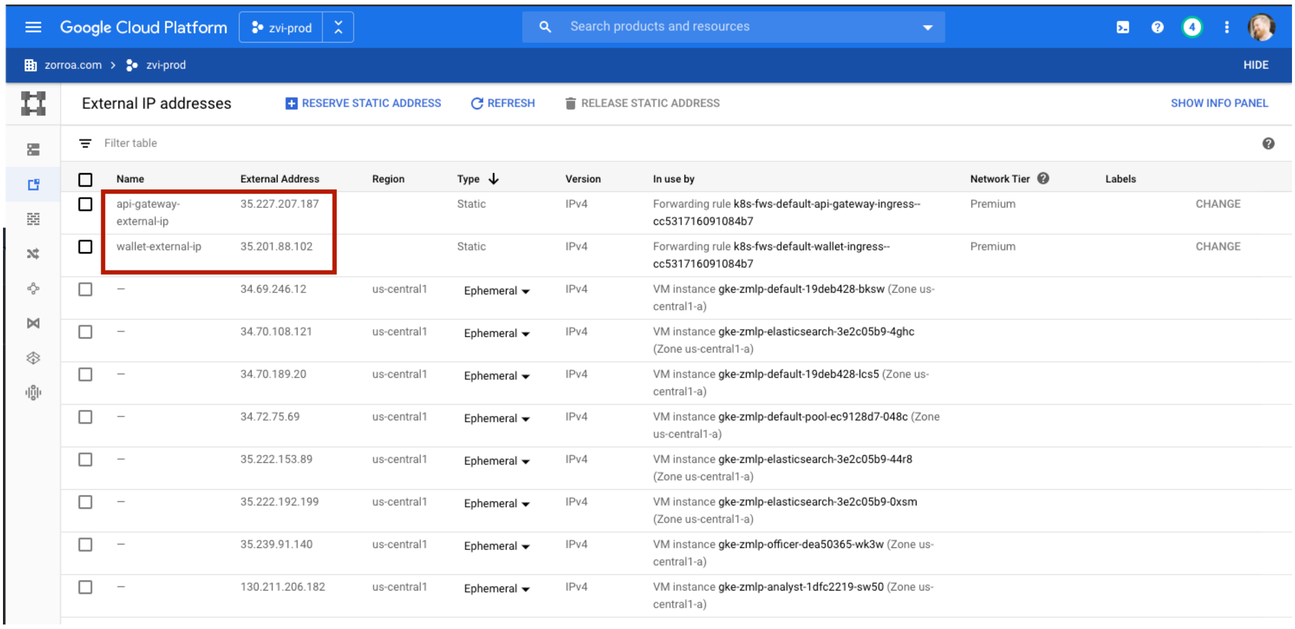
Get the IP Addresses

During the previous step two external IP addresses were reserved in the GCP project. We'll need to find those and jot them in order to configure the DNS for the domains.

Go to the GCP Console tool below to find the IP addresses. Write down the IP addresses associated with `api-gateway-external-ip` and `wallet-external-ip` .

https://console.cloud.google.com/networking/addresses/list https://console.cloud.google.com/networking/addresses/list

When go to the GCP Console double-check that you are in the correct GCP project.



☐	Name	External Address	Region	Type	Version	In use by	Network Tier	Labels	
☐	api-gateway-external-ip	35.227.207.187		Static	IPv4	Forwarding rule k8s-fws-default-api-gateway-ingress--cc531716091084b7	Premium		CHANGE
☐	wallet-external-ip	35.201.88.102		Static	IPv4	Forwarding rule k8s-fws-default-wallet-ingress--cc531716091084b7	Premium		CHANGE
☐	–	34.69.246.12	us-central1	Ephemeral ▾	IPv4	VM instance gke-zmlp-default-19deb428-bksw (Zone us-central1-a)			
☐	–	34.70.108.121	us-central1	Ephemeral ▾	IPv4	VM instance gke-zmlp-elasticsearch-3e2c05b9-4ghc (Zone us-central1-a)			
☐	–	34.70.189.20	us-central1	Ephemeral ▾	IPv4	VM instance gke-zmlp-default-19deb428-lcs5 (Zone us-central1-a)			
☐	–	34.72.75.69	us-central1	Ephemeral ▾	IPv4	VM instance gke-zmlp-default-pool-ec9128d7-048c (Zone us-central1-a)			
☐	–	35.222.153.89	us-central1	Ephemeral ▾	IPv4	VM instance gke-zmlp-elasticsearch-3e2c05b9-44r8 (Zone us-central1-a)			
☐	–	35.222.192.199	us-central1	Ephemeral ▾	IPv4	VM instance gke-zmlp-elasticsearch-3e2c05b9-0xsm (Zone us-central1-a)			
☐	–	35.239.91.140	us-central1	Ephemeral ▾	IPv4	VM instance gke-zmlp-officer-dea50365-wk3w (Zone us-central1-a)			
☐	–	130.211.206.182	us-central1	Ephemeral ▾	IPv4	VM instance gke-zmlp-analyst-1dfc2219-sw50 (Zone us-central1-a)			

Example image of the two IP addresses to record.

Add A Records to the DNS

Now you'll need to add an A record to the DNS for each of the ip addresses. If this is an internal deployment hosted at boonai.app you can contact devops.

The values to use for the A Records are below:

Domain	IP Address
Boon AI Console Domain	wallet-external-ip address
Boon AI REST API Domain	api-gateway-external-ip address

Change Default Admin Password

By default there is an admin user with the credentials below. YOU MUST CHANGE THIS PASSWORD IMMEDIATELY.

software@zorroa.com/admin