Vernal

Team 25 Product Backlog

Winston Ngo, Dennis Pham, Hpung San Awng, Aaron Ni

Problem Statement

One of the most time-consuming tasks many organizations and developer teams face is having to deploy, scale, and manage applications in the cloud. Many teams find themselves having to sink a considerable amount of time into handling deployments and building the pipelines necessary to get their app into the cloud.

Background Information

Audience

These days with the software development process getting increasingly more complicated and with new tech stacks and technologies releasing regularly, it is more important than ever for developers to be able to focus on the development of their product rather than having to set up complicated pipelines just to deploy their apps. Vernal is designed with developers in mind, allowing them to seamlessly deploy their app from their own repository in just a few clicks.

Similar Platforms

There are a number of existing services today that help developers quickly get their app into the cloud. Products like AWS Amplify and Heroku exist to allow developers to deploy their full-stack applications without needing to manage their own infrastructure. Other services like Netlify and Vercel also exist, but are only limited to serverless solutions.

Limitations

Although there are numerous existing services out there that let you quickly deploy your full-stack applications to the web, they also lack many other features that busy developers and organizations need. Vernal provides the ability to perform tasks such as automated security and vulnerability scans, which is not a feature included in many existing solutions. Along with this, Vernal also includes the ability to enable OIDC authentication for their application with just a few clicks, saving developers considerable time from having to implement their own solution for authentication. Developers can also view detailed metrics and monitor the performance of their deployed applications.

Vernal seeks to provide all of these features together in an easy-to-use, cohesive developer platform.

Functional Requirements

1. As a developer, I would like to be able to register for a Vernal account.
2. As a developer, I would like to be able to login to my Vernal account.
3. As a developer, I would like to reset my password if I forget it.
4. As a developer, I would like to be able to manage deployments and configurations.
5. As a developer, I would like to easily deploy my application from my repository to the cloud.
6. As a developer, I would like to provide environment variables for my hosted applications.
7. As a developer, I would like to deploy a PostgreSQL database and connect it to my applications.
8. As a developer, I would like to see all of the web apps created in my organization so that I can collaborate on other projects.
9. As a developer, I would like to be able to start an app from a Next.js/FastAPI template.
10. As a developer, I would like to be able to start an app from a Next.js/Express template.
11. As a developer, I would like to be able to start an app from a Next.js/Apollo GraphQL template.
12. As a developer, I would like to be able to start an app from a Next.js/TypeGraphQL template.
13. As a developer, I would like my application's health to be monitored and for the deployment to be restarted if errors occur.
14. As a developer, I would like insights from Lighthouse audits to improve performance, accessibility and SEO.
15. As a developer, I would like to be able to view my application's live topology in the Kubernetes cluster.
16. As a developer, I would like to visualize CPU usage for my app.
17. As a developer, I would like to visualize memory usage for my app.
18. As a developer, I would like to visualize network traffic for my web app.
19. As a developer, I would like to see the logs from my app.

20. As a developer, I would like to see errors from my app.
21. As a developer, I would like to be able to view CI/CD logs.
22. As a developer, I would like the ability to automatically scale my applications for more demand.
23. As a developer, I would like to use a cohesive dashboard for my deployment and scaling needs all in one place.
24. As a developer, I would like to be able to navigate within a consistent and user-friendly interface.
25. As a developer, I would like to see documentation for each app.
26. As a developer, I would like to enable OIDC authentication for my application to secure user access.
27. As a developer, I would like to utilize the automated security and vulnerability scans proved by Vernal to maintain security integrity of my applications.
28. As an organization, we would like accounts to be stored in a secure database so that only authorized users can use the app.
29. As an organization, we would like our developers to follow guidelines for creating documentation so that all documentation is consistent.
30. As an organization, we would like our developers to be able to document their code on dedicated documentation sites.
31. As an organization, we would like to centrally manage our dev team's cloud deployments through Vernal to ensure consistency and efficiency in our deployment process.
32. As an organization, we would like our teams to post messages to a community board.
33. As an organization, we would like our teams to be able to reply to messages in a community board.
34. As an organization, we would like our teams to be able to like and dislike messages.
35. As a developer, I would like to be able to edit my messages.
36. As a developer, I would like to be able to edit my profile to add a picture and description.
37. As a developer, I would like to receive alerts whenever a new vulnerability is detected, so that I can take action as quickly as possible (if time allows).
38. As an organization, we would like our teams to be able to have a shared knowledge hub so our developers can expand their skills (if time allows).
39. As a developer, I would like to be able to create my own or import a template for app deployment (if time allows).

Non-Functional Requirements

Architecture

There are 5 main parts of our application architecture: the frontend web UI, backend REST API, PostgreSQL database, Redis, and the Kubernetes cluster(s) to power the entire platform.

The backend API will be written in TypeScript using Backstage, with the underlying web framework being Express. Knex is the SQL query builder used for the database and Keyv is used for access to Redis for caching.

The frontend will also be written in TypeScript using Backstage, with the underlying web framework being React. It will send requests to our API to fetch and update data, as well as execute actions from the user.

The Kubernetes cluster(s) will be the key to powering this entire application.

- Argo CD is used to declaratively manage Kubernetes resources
- Crossplane is used to declaratively manage cloud provider resources
- Istio is a service mesh that enables us to secure, control, and manage network traffic between services
- Kiali provides graph visualizations for operators to monitor and observe network traffic
- Loki collects, aggregates, and stores logs from applications
- Prometheus, Node Exporter, and Kube State Metrics collect a multitude of different metrics about the clusters and applications/services hosted on them
- Grafana is an interactive visualization dashboard that provides alerts, charts, and graphs for collected metrics
- Sealed Secrets is a tool to encrypt secret values and store them declaratively in a git repository (goes hand in hand with Argo CD)

Security

As with all applications, security is crucial to the success of the Vernal. Not only will it be deploying and hosting production applications for our users, but it will also have access to the git repositories where their source code is hosted. On the application side, Backstage already has many built-in features to prevent common security vulnerabilities, such as CSRF prevention. Additionally, we will have RBAC for resources in the application and authentication implemented via OIDC with Keycloak.

For the one-click authentication for hosted applications, we will have CSRF prevention built-in so that developers do not have to worry about implementing it in their backend code, allowing them to focus on their application's core business logic.

Usability

Developers should spend more time working on their apps and less time on configuring them. With this in mind, Vernal should be very simple to use so developers can quickly navigate the UI to easily build and deploy a project. With simple dropdowns, text fields, and selection menus, setting up projects should be very simple for both new and seasoned developers. If time allows, we would like our platform to be a central hub for developers where we would include social and communication features.

Deployment

As Vernal is a cloud-agnostic solution, it is deployable to any cloud platform with Kubernetes support. However, we plan to host our instance on AKS.