PREvant (Preview Servant): Composing Microservices into Reviewable and Testable Applications

Marc Schreiber, M. Sc.

aixigo AG, Aachen, Germany
marc.schreiber@aixigo.de

Currently, a trend is for increasing numbers of enterprises to develop microservices to build their applications [1], because microservices are scalable and have fast deployment cycles, superior quality, and greater flexibility [3, 4]. Furthermore, numerous companies migrate their on-premise applications to microservice architectures [2] because of the abovementioned advantages in combination with agile software development. However, when development teams begin to develop microservices, they can face major challenges related to quality assurance caused by distributed feature development.

First, microservices must be deployed on an infrastructure to perform automatic and user-based acceptance tests, thereby ensuring feature correctness. Developers often use common containerization techniques to package and deploy their microservices. However, developers must manage complexity of deployment setups when they deploy the whole application, which consists of multiple microservices distributed across numerous source code repositories. To deploy their services, developers must create complex continuous delivery pipelines that utilize container orchestration platforms, such as Kubernetes.

Furthermore, additional challenges are generated by composing microservices into a single application that follows strict branching workflows, such as git flow. For example, if developers want to provide domain experts with a preview of a new feature of a microservice, they must create a complex continuous delivery pipeline to automatically deploy their new features and all remaining services as one application for review purposes. Often, this setup is too complex, and features are made available to the experts only when they have been merged into the mainline branch, making feature-based previews impossible.

To solve these challenges, this paper introduces *PREvant (preview servant)*, a software tool which provides a simple RESTful API for deploying and composing containerized microservices as reviewable applications, as illustrated in Figure 1. PREvant's API serves as a connector between the continuous delivery pipelines (see *Build, Package, Deploy* in Figure 1) of all involved microservices and the infrastructure that hosts the applications.

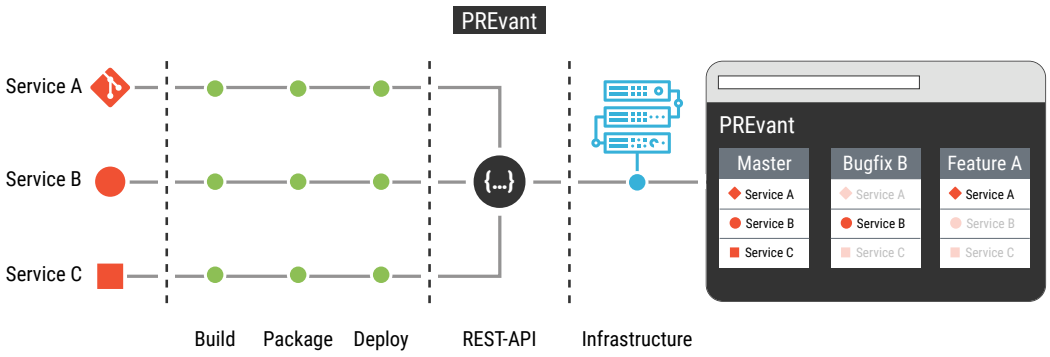

Figure 1: Composing Microservices with PREvant into Reviewable Applications

When PREvant is up and running, developers are only required to extend their continuous integration pipelines with two stages. In addition to the well-known stages, developers require *Package* and *Deploy* stages. In the *Package* stage, the application is packaged as a container image and pushed to a container image registry (e.g. a Docker registry). Furthermore, the *Deploy* stage makes a simple REST call to PREvant's API, which instructs PREvant to deploy the container image to an infrastructure.[1] When each microservice's mainline branch (e.g. branch *master*) contains the build pipeline configuration of the *Package* and *Deploy* stages and each microservice has been deployed, PREvant composes all master branch versions of the microservices into an application for review. Additionally, PREvant provides a web interface to the services, which can be used by domain experts to review the application as a whole (see *Master* in Figure 1).

The master application serves as a template for all new features and bugfixes of the application that must be reviewed. When developers create a new feature or bugfix branch, PREvant's API can be used to deploy the new version of the specific microservice. To provide a fully functional application that can be reviewed by the domain experts, PREvant replicates all remaining services from the master application. With all microservices composed into one reviewable application (e.g. *Feature A*), domain experts can review changes before the feature is merged into the mainline branch. Additionally, PREvant provides a web interface to access the reviewable applications that features issue-tracking integration, versioning information, and Swagger UI integration.

Through PREvant's simple abstraction of deployment, developers, team managers, domain experts, and sale managers can benefit from the following advantages:

- The RESTful API provides easy-to-use deployment and composition of microservices, which enables domain experts to review new features and bugfixes.
- The web interface provides an integrated view of all components to be tested (services and their APIs and frontends). Furthermore, it supports the mantra of "getting things done" through the integration of issue-tracking systems; domain experts, developers, and managers can test and review the application's developments as soon as possible, which supports agile software development approaches.
- PREvant has a simple setup mechanism integrated into the web interface allowing dedicated previews in minutes for demonstrating the application to potential customers.
- PREvant is available as open-source software at `http://github.com/aixigo/PREvant`.

References

[1] Anne Marie Glen. *[DZone Research] Microservices Priorities and Trends*. DZone, Inc. July 2018. URL: https://dzone.com/articles/dzone-research-microservices-priorities-and-trends (visited on 11/21/2018).

[2] Pooyan Jamshidi, Aakash Ahmad, and Claus Pahl. "Cloud Migration Research: A Systematic Review". In: *IEEE Transactions on Cloud Computing* 1.2 (2013), pp. 142–157. DOI: 10.1109/tcc.2013.10.

[3] Cesar Saavedra. *The State of Microservices Survey 2017 – Eight trends you need to know*. Red Hat, Inc. Dec. 2017. URL: https://middlewareblog.redhat.com/2017/12/05/the-state-of-microservices-survey-2017-eight-trends-you-need-to-know/ (visited on 11/21/2018).

[4] Markos Viggiato et al. *Microservices in Practice: A Survey Study*. 2018. arXiv: 1808.04836.

[1] By design, PREvant will support various infrastructures such as Docker, Kubernetes, and Mesos because it abstracts them.