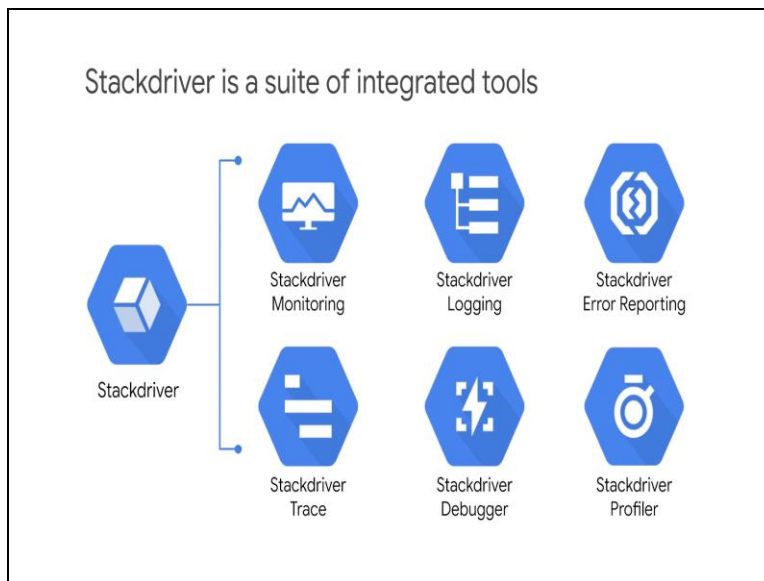


**Google Cloud Computing Foundation Course**  
**Seth Vargo**  
**Google Cloud**

**Lecture-66**  
**Stackdriver**

In the previous topic you saw that certain key activities need to be performed to monitor and manage existing services applications and infrastructure. In this topic you will learn how stack driver can be used for monitoring, logging error reporting tracing and debugging your applications in the cloud.

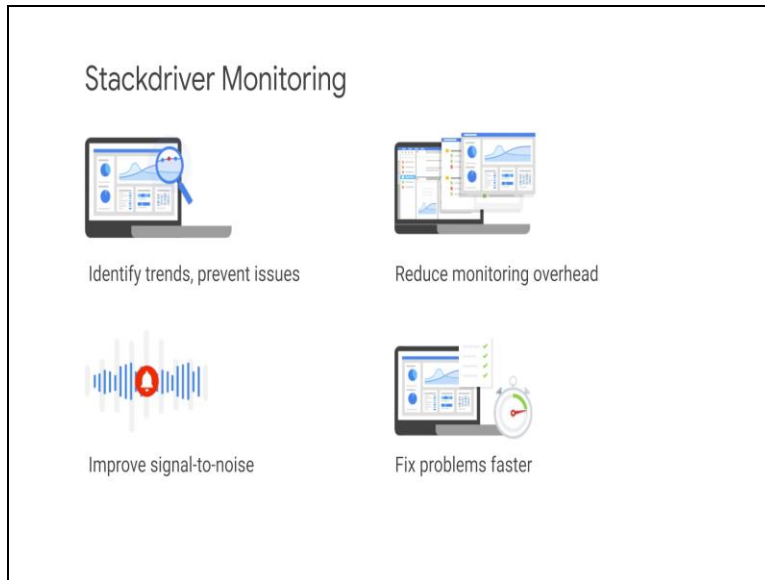
**(Refer Slide Time: 00:21)**



Stackdriver provides powerful monitoring, logging and diagnostics for applications on Google cloud platform. It equips you with insight into the health performance and availability of cloud powered applications enabling you to find and fix issues faster. Stackdriver gives you access to many different kinds of signals for your infrastructure platforms, virtual machines, containers middleware and all your application tiers including logs, metrics and traces.

It gives you insight into your applications health, performance and availability. So, if issues occur you can fix them faster.

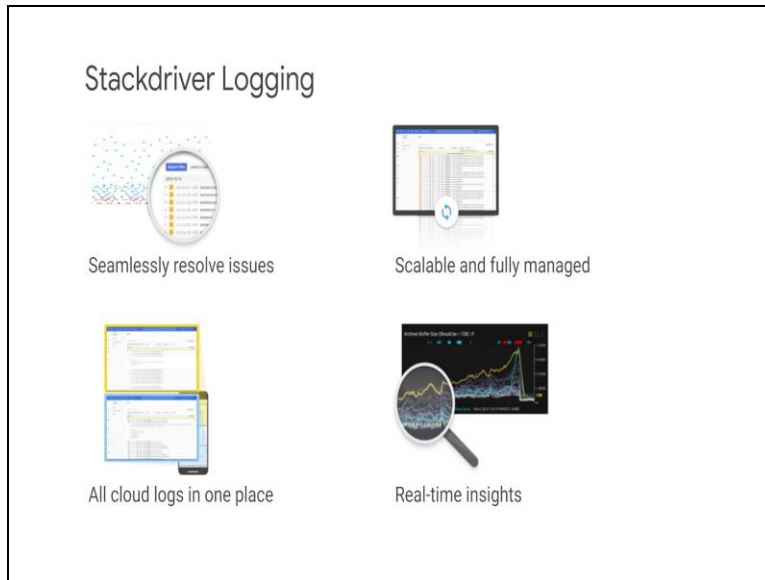
**(Refer Slide Time: 01:00)**



Let us start by looking at stack driver monitoring a full stack monitoring service that discovers and monitors cloud resources automatically. Flexible dashboards and rich visualization tools help you identify emerging issues. Anomaly reporting pattern detection and exhaustion prediction provide insights into longer-term trends that may require attention. Monitoring provides a single integrated service for metrics, dashboards, uptime monitoring and alerting. This means that you spend less time maintaining disparate systems. Advanced alerting capabilities including rate of change, cluster aggregation and multi condition policies help ensure you are notified when critical issues occur while reducing the likelihood of a false positive.

Integrated uptime monitoring and health checks ensure quick notification of failures it is possible to drill down from the alerts dashboard to logs and traces in order to identify the root cause of a problem quickly.

**(Refer Slide Time: 02:06)**

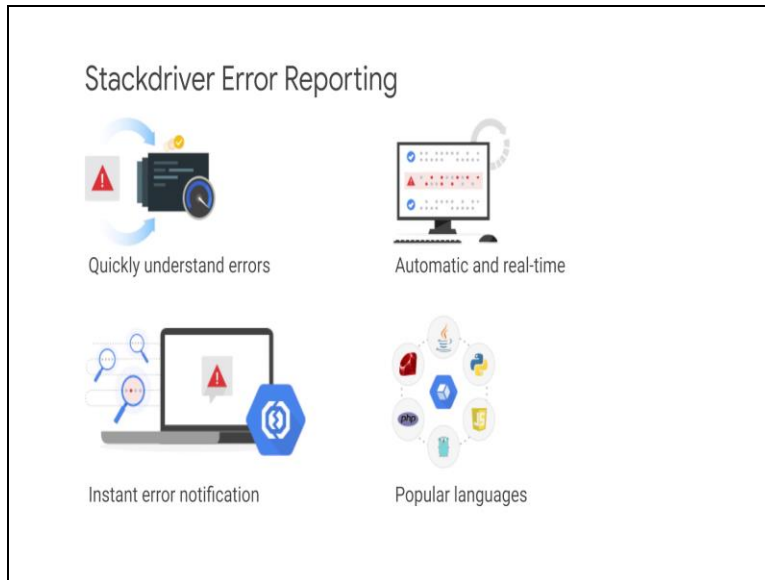


Stackdriver logging is a real-time log management and analysis service. Stack driver logging is a fully integrated solution that works seamlessly with stackdriver monitoring, stackdriver trace, stack driver error reporting and stack driver debugger. The integration allows users to navigate between incidents, charts, traces, errors and logs. This helps users quickly find the root cause of issues in their applications and systems.

Logging is built to scale and works well it is sub-second ingestion latency even at terabytes per second. Logging is a fully managed solution that takes away the overhead of deploying or managing a cluster thus allowing you to focus your energy on innovation and building your product. Logging provides a central place for all your logs. You can also configure stackdriver to export logs to other systems automatically.

Stackdriver logging allows you to analyze high-volume application and system level logs in real time. Advanced log analysis can be achieved by combining the power of the stack drivers suite with the data and analytics products of Google cloud platform. For example you can create powerful real-time metrics from the log data and analyze that log data in a real-time using a tool like BigQuery.

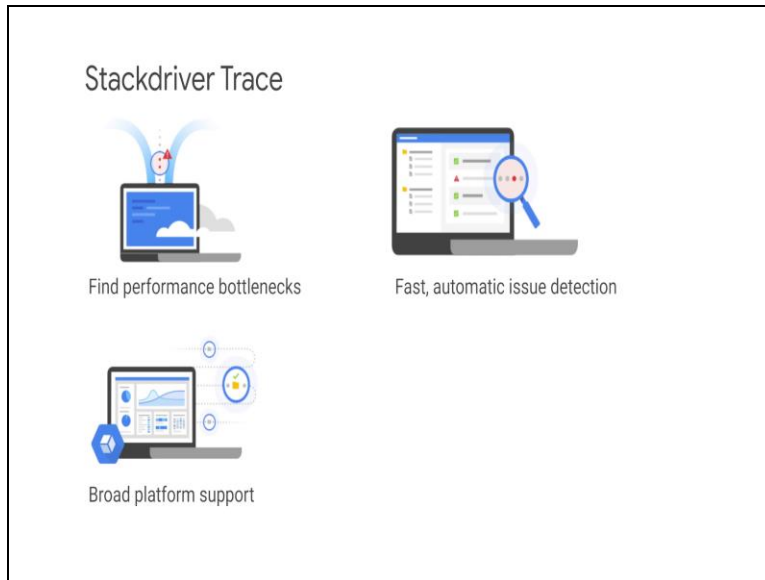
**(Refer Slide Time: 03:31)**



Stackdriver error reporting allows you to identify and understand application errors through real-time exception monitoring and alerting. Error reporting also allows you to see your applications top errors in a single dashboard. Real production problems can often be hidden across mountains of data. Stackdriver error reporting helps you see problems through the noise by constantly analyzing exceptions and intelligently aggregating them into meaningful groups that are tailored to your programming language or framework.

Stackdriver error reporting is constantly watching your service and instantly alerts you when a new application error cannot be grouped together with the existing ones. You can directly jump from a notification to the details of a new error. The exception stack trace parser is able to process Go, Java, Dotnet, Nodejs, PHP, Python and Ruby. You can also use Google client libraries and REST API's to send errors with stackdriver logging.

**(Refer Slide Time: 04:36)**

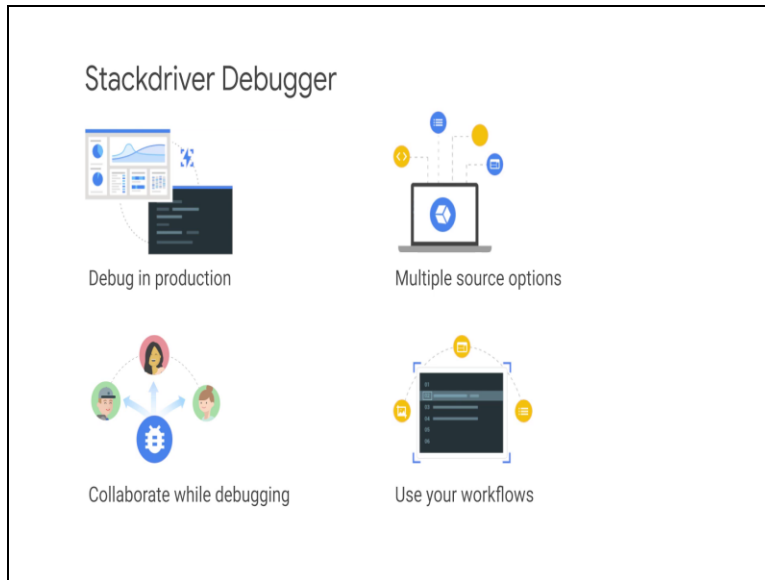


Stackdriver trace is a distributed tracing system that collects latency data from applications and displays it in the Google Cloud console. Using stackdriver trace you can expect detailed latency information for just a single request or with aggregate latency across your entire application. You can quickly find where bottlenecks are occurring and more quickly identify their root causes. Stackdriver trace continuously gathers and analyzes data from applications to automatically identify changes to an applications performance.

These latency distributions are available through the analysis reports feature. They can be compared over time or versions and stackdriver trace will automatically generate an alert if it detects a significant shift in an applications latency profile. The language-specific SDKs of stackdriver trace can analyze projects running on virtual machines. The stackdriver trace SDK is currently available for Java, Nodejs, Ruby and Go.

And the stackdriver trace API can be used to submit and retrieve trace data from any source or any language. A zip can collector is also available which allows Zipkin tracers to submit data to stackdriver trace. Stackdriver trace works out of the box on many GCP services including App Engine.

**(Refer Slide Time: 06:00)**

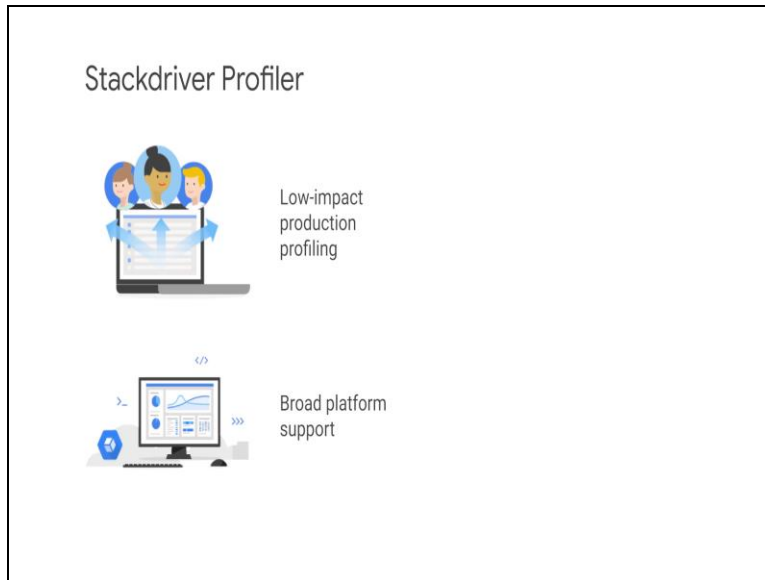


Stack driver debugger is a feature of Google cloud that lets you inspect the state of a running application in real time without stopping it or slowing it down. Stackdriver debugger can be used with production applications with just a few mouse clicks you can take a snapshot of your running application state or inject a new logging statement that snapshot captures the call stack and variables at a specific code location the first time any instance execute that code.

The injected log point behaves as if it were part of the deployed code writing the log messages to the same log stream. Stackdriver debugger is easy when the source code is available to Google Cloud. It knows how to display the correct version of the source code when a version control system such as Google Cloud source repositories get hub, bitbucket or get lab are available. Users can easily collaborate with other teammates by sharing their debug sessions.

Sharing a debug session in stackdriver debugger is as easy as sending the console URL. Stackdriver debugger is integrated into the existing developer workflows you are familiar with. You can launch stackdriver debugger and take snapshots directly from stackdriver logging, error reporting, dashboards, integrated development environments and even the G-cloud command line.

**(Refer Slide Time: 07:22)**



Interface poorly performing code increases the latency and cost of applications and web services everyday. Stackdriver profiler continuously analyzes the performance of CPU or memory intensive functions that are executed across your applications. While it is possible to measure code performance and development environments the results generally do not map well so what is happening in production. Unfortunately many production profiling techniques either slow down code execution or can only inspect a small subset of the codebase.

Stackdriver profiler uses a statistical technique an extremely low impact instrumentation that runs across all production application instances to provide a complete picture of an applications performance without slowing it down. Stack driver profiler allows developers to analyze applications running anywhere including GCP and other cloud platforms on-premise or with the support for Java, Go, Nodejs' and Python.