Nomad

Extending Nomad with Plugins

HashiCorp

HashiCorp Nomad

- Workload orchestrator: maps workload to compute resources.

- Designed to run **any workload on any platform**.

- Runs on Linux, Windows, or macOS.

- Runs Dockerized or legacy applications.

- Extensible runtimes via **task driver plugins.**

- Extensible hardware support using **device plugins.**

Chris Baker, PhD

Senior Engineer, **Nomad Ecosystem Team**
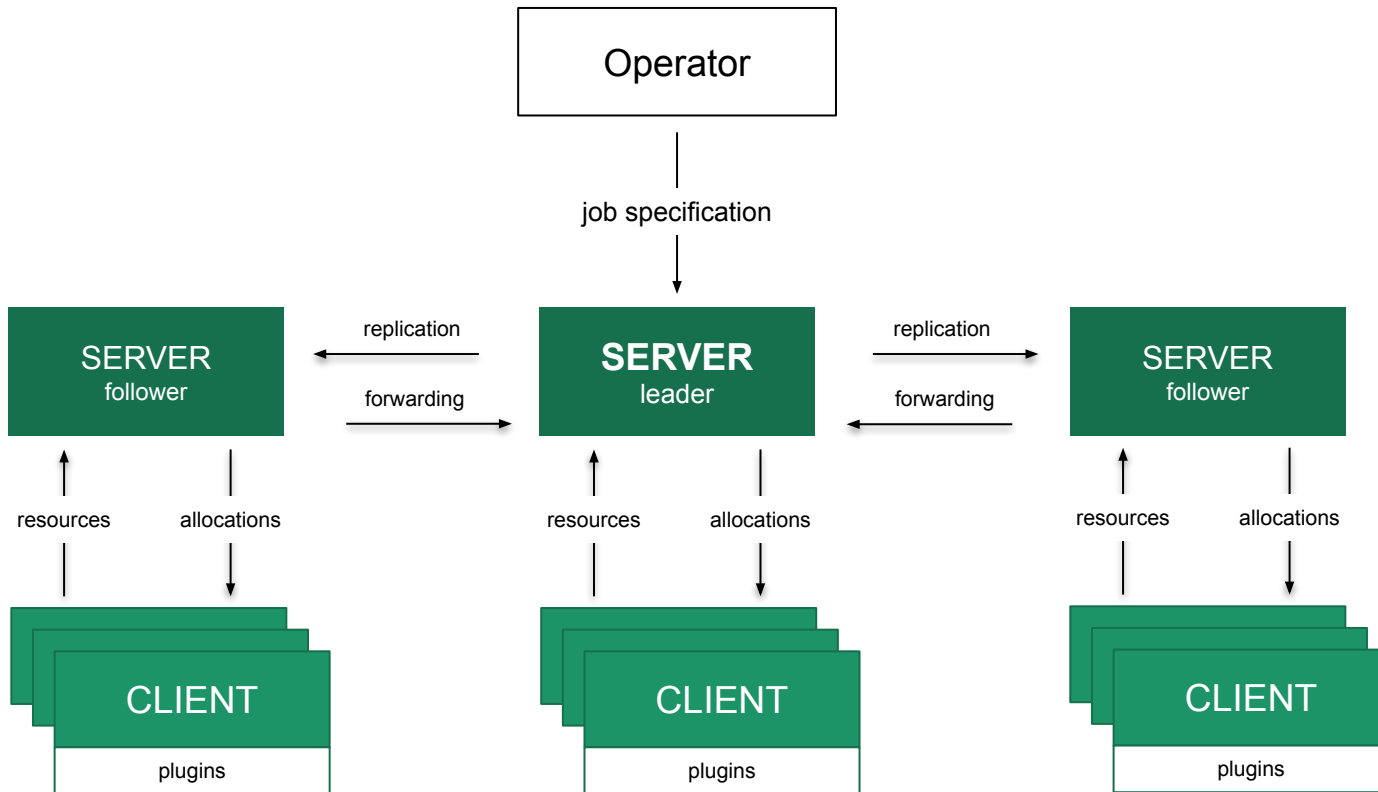
he/him

HashiCorp

Outline

Demo

Nomad Plugin Architecture
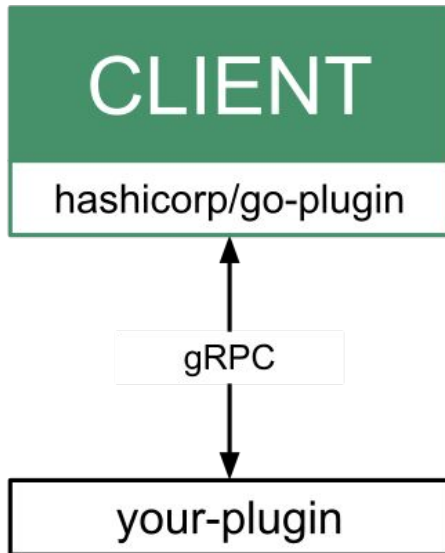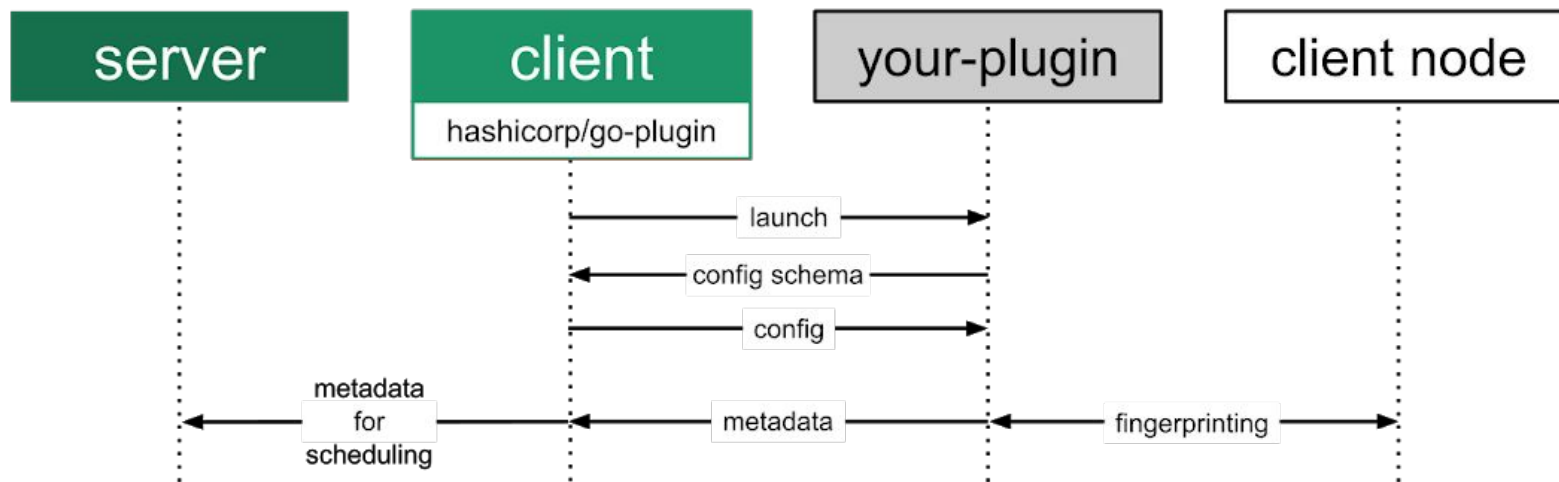
Example Plugins

Nomad Cluster Architecture

Nomad Plugin Architecture

```
> nomad agent -plugin-dir=plugins …
```

Nomad Plugin Architecture

Nomad Device Plugins
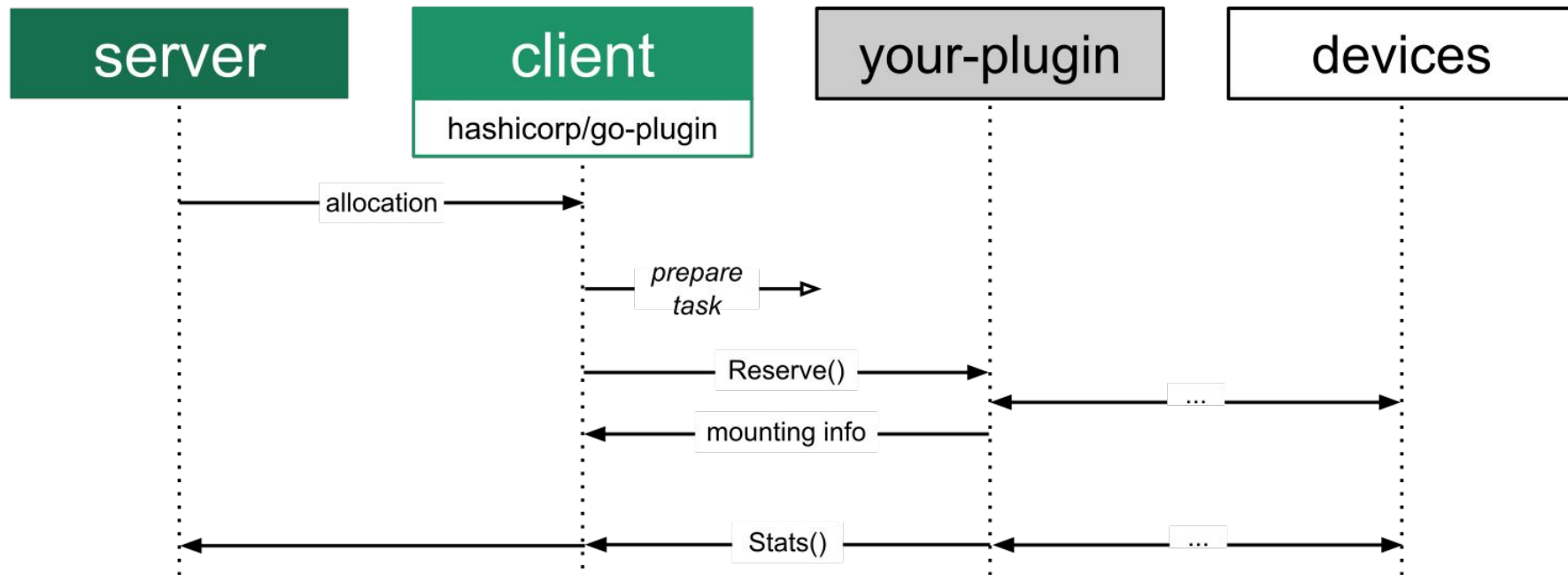
Device Plugin Interface

```go
type DevicePlugin interface {

    Fingerprint(ctx context.Context) (<-chan *FingerprintResponse, error)

    Reserve(deviceIDs []string) (*ContainerReservation, error)

    Stats(ctx context.Context, int time.Duration) (<-chan *StatsResponse, error)

}
```
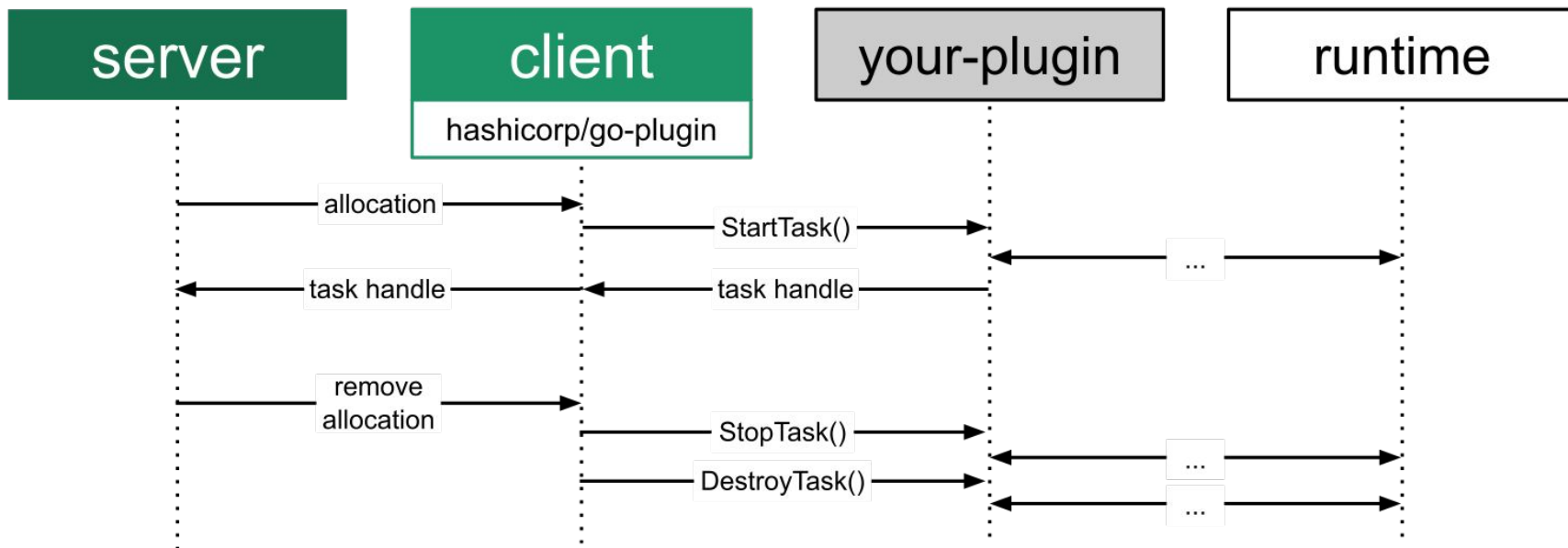
Nomad Task Driver Plugins

Driver Plugin Interface

```go
type DriverPlugin interface {
        StartTask(*TaskConfig) (*TaskHandle, *DriverNetwork, error)
        RecoverTask(*TaskHandle) error
        WaitTask(...) (<-chan *ExitResult, error)

        StopTask(...) error
        DestroyTask(...) error

        SignalTask(taskID string, signal string) error
        ExecTask(...) (*ExecTaskResult, error)
}
```

Demo Code Review

```go
// writeFingerprintToChannel collects fingerprint info, partitions devices into
// device groups, and sends the data over the provided channel.
func (d *SkeletonDevicePlugin) writeFingerprintToChannel(devices chan<- *device.FingerprintResponse) {

        usbCtx := gousb.NewContext()
        defer usbCtx.Close()

        // OpenDevices is used to find the devices to open.
        d.logger.Info("fingerprinting USB devices")
        foundDevs, err := usbCtx.OpenDevices(func(desc *gousb.DeviceDesc) bool {
                // The usbid package can be used to print out human readable information.
                d.logger.Info("examining device", "vendor_id", desc.Vendor, "product_id", desc.Product)
                return uint16(desc.Vendor) == d.targetVendorId && uint16(desc.Product) == d.targetProductId
        })

        if err != nil {
                d.logger.Warn("error opening usb devices", "err", err)
                devices <- &device.FingerprintResponse{
                        Devices: nil,
                        Error:   err,
                }
                return
        }

        d.deviceLock.Lock()
        defer d.deviceLock.Unlock()
```

Conclusion

Extensibility through Plugins

Support for scheduling **any workload** against **any hardware infrastructure**.

Growing Ecosystem of Plugins

Numerous internal and external plugins to use as examples.

Future Plugin Systems

Future versions of Nomad will use plugins for further extensibility.

Nomad

Demo available on GitHub:

https://github.com/cgbaker/hashitalk-2020-nomad-plugins

cgbaker / hashitalk-2020-nomad-plugins

⊙ Unwatch ▾ | 1 | ★ Sta

generated from hashicorp/nomad-skeleton-driver-plugin

<> Code   ⓘ Issues 1   ⑂ Pull requests 0   ▶ Actions   ▥ Projects 0   ▤ Wiki   ⛉ Security   ⊩ Insights   ⚙ Setti

No description, website, or topics provided.

Manage topics

⟳ 5 commits   ⑂ 1 branch   ▥ 0 packages   ◌ 0 releases   👥 1 contributor

Branch: master ▾   New pull request   Create new file   Upload files   Find file

cgbaker added usb device demo   Latest con

📁 example-device	updated README, added top-level makefile, moved example config/job into
📁 example	added usb device demo
📁 python-driver	Initial commit
📄 .gitignore	Initial commit
📄 GNUmakefile	updated README, added top-level makefile, moved example config/job into

Thank You

hello@hashicorp.com
www.hashicorp.com