

[RFC] Billing: Access all product tiers without credit card for internal users

Summary: Allow internal users to have access to all product tiers of company services in production without having to enter a credit card number

This Request for Comment (RFC) aims to describe how we can architect the billing system to reduce friction points for internal users. This covers both internal users not accruing charges and being able to access all tiers of products in production without being required to enter a personal credit card number.

Background

The company's Cloud Portal UI is used by customers, but it is also used by a variety of internal users across four different environments: local dev, remote dev, integration, and production. There are two primary friction points where the billing system as developed for customers may interfere with internal use and test cases: accruing usage costs, and which product tiers are available to select in the UI. At present, unless the user has either a credit card added or is registered with a contract, they only have access to the development tier. Thus, one cannot spin up small, medium, or large clusters without a credit card. Additional tiers and cluster sizes will be added in the future. And additional tier specific features are planned as well. This highlights the importance of internal users being able to access, use, and test all tiers and features.

A billing account may have no payment method associated. If it has credits remaining, then it would be in a TRIAL state. If no credits remain, then it would be in an INCOMPLETE state. To be ACTIVE a billing account must have one or more payment methods associated with the account. Credit card information can be added which is called an On Demand payment method. Or the billing account can be assigned to a contract. Payment method can be added at billing account creation or on update. Additional payment methods such as flexible consumption are under development.

Account Status	Payment method Added?	Trial Credits Remaining?
Trial	No	Yes
Active	Yes	Yes or No
Incomplete	No	No

Currently a LaunchDarkly feature flag (`BillingOptOut`) is utilized to opt an org out of accruing usage charges. This does not stop usage data being collected by the cloud-metering service but that account's invoices will automatically be marked as paid and the account's actions will always be allowed by the billing service. There is currently a Google form to request this opt out. A Product Manager then adds the user to the user segment used by the feature flag.

For local dev, remote dev, and integration environments there are test Stripe credit card numbers that can be used to gain access to all tiers, but test card numbers cannot be used in production.

Work was recently completed to indicate whether an org is employee owned via the `employee_owned` field on the `getOrg` endpoint and this will also be leveraged to determine if an organization is owned by an internal user.

There are a number of reasons internal users may need access to all tiers in the production environment. For example, Sales may need it for demos and sample architectures. Support needs access for testing, troubleshooting and reproducing problems. Education and marketing may need it for product training and enablement. Internal users outside of engineering will not have access to dev and int environments and so production is their only option.

Proposal

Give internal users the ability to utilize either payment method, On Demand (*i.e.*, credit card) or contract, without requiring real credit card information and without contaminating data about real customer contracts.

Adding designations for `internal_onDemand` and `internal_contract` to the billing account will let us explicitly represent these statuses while giving internal users the most options in terms of functionality.

Buttons to set Internal OnDemand or Internal Contract will be added to the Billing Accounts page of the Admin UI. Additionally Internal OnDemand will automatically be set to true in production for orgs designated as `employee_owned` (which generally means they have a `@company.com` email). OnDemand is currently the default experience for HCP. We will only set this in production because there are significant developer preferences that would need to be considered for the other environments. See Implementation section for more details.

Contracts will not actually be created for Internal Contract, thereby reducing bloat and noise to the contracts table used to track actual customer contracts.

Type	Where status is stored	Usage calculated by metering-service	View of billing summary	Stripe invoices generated	How it's set
Internal Contract	Internal_contract true for the users' billing account	yes	no	no	Via Admin UI
External Contract	Row in Contracts table associated to the users billing account; contract salesforce_id not null	yes	no	no	Via Admin UI
Internal On Demand	Internal_onDemand true for the users' billing account	yes	yes	no	Set automatically for where org designated employee_owned, can also be set via Admin UI
External On Demand	User has credit card associated with their account (could be fake stripe test credit cards for environments other than production)	Yes	yes	yes	Via adding credit card information in HCP

Validations will be added to ensure that Internal OnDemand and Internal Contract are not both true at the same time, primarily for clarity of user experience. This is not a serious concern as currently a user can have both a credit card (*i.e.*, an On Demand payment method) and a contract. In this case the contract is the primary payment method and the credit card is a fallback if the contract expires or is removed.

In order to test trial status, a user can always sign up for a new account as TRIAL is the default starting state for all accounts. This implementation allows users to go from TRIAL status to ACTIVE status (for either On Demand or Contract) or to go from ACTIVE status of one payment method to the other. INCOMPLETE and DELINQUENT states/transition between states are out

of scope for this RFC but will likely require thought in the future for additional use cases. DELETED and DELETING statuses are not addressed and not impacted by this proposal.

Abandoned Ideas

Require Personal Credit Card

One solution is requiring users to add a personal credit card through the Cloud Portal UI and then add them to the feature flag or the `employee_owned` designation to ensure they don't get billed. The downsides to this include not every employee may have a personal credit card. Additionally, if the user leaves the company and is removed from the feature flag/no longer designated as an employee, then they would get charged for usage. And if they signed up for the account with their company email, they may not be able to access the account to delete the resources. Which might mean their only option would be to create a Support ticket. This creates a lot of complexity and potential stress for employees as it would be easy to have several hundred dollars a month in charges that you fear could unexpectedly land on your personal card.

Put internal users on a contract for their payment method

Currently users with Admin UI access can register themselves with a contract to avoid having to enter a credit card number. Internal users could be subscribed to contracts so they would not have to enter a credit card number. Contracts are still very early in development, and we anticipate building out additional features to track things like overages (usage beyond what is allowed on the specific type of contract). As a result, once those features are developed internal users would still risk getting charged if they use more than allowed under the contract.

For the purpose of marketing metrics, we are currently describing as internal any contracts without a salesforceID. This approach would also create contracts in our database that are noisy and not useful. This would also limit internal users to the contract UI experience and not allow them to see the billing summary page nor how cost/usage accumulates.

Implementation

Add 2 new boolean columns, `internal_onDemand` and `internal_contract` to the BillingAccount table and model. These columns will be used to stop users from being charged in the same way the feature flag currently does, and to allow users access to all tiers of packages (the way adding a credit card or registering for a contract currently does).

In production, `getOrg` will be queried and if the organization is employee owned then `internal_onDemand` will automatically be set to true. We will only do this in production for now because it's not clear what the default should be in development and integration (where users need to test trial and contract and invoicing and all functionality). We would need greater clarity on the preferred developer experience. On Demand is the general default experience, as

a contract is something customers must negotiate with sales and cannot do self-service, and therefore it makes sense to use that as the default.

All organizations currently in the `BillingOptOut` flag will be set to `internal_onDemand true` and then the feature flag can be removed. See the section on “Migration of Existing Internal Users” for more details.

A contract will not actually be created where `internal_contract` is true, thereby reducing the amount of internal/non-customer contracts saved unnecessarily in the database. We will still utilize a real contract in some use cases such as contract penetration testers where marking them as internal users feels inappropriate.

The new internal payment fields also need to be added to the Protobuf private requests `CreateBillingAccountRequest` and `UpdateBillingAccountRequest`. The ability to set these fields needs to be exposed in the private endpoints in order to allow it to be set via the Admin UI, for example in the case of existing accounts where the user signed up with a personal email via GitHub or where the user wants to toggle between payment methods.

```
// CreateBillingAccountRequest is the request to add a new Billing Account for
// an organization.
```

```
type CreateBillingAccountRequest struct {
    // id is the user-settable ID for this Billing Account. It must be unique
    // within the organization.
    Id string

    // organization_id is the ID of the organization to which the
    // Billing Account belongs.
    OrganizationId string

    // name is the Billing Account's user-defined name.
    Name string

    // project_ids is the list of IDs of all projects associated with the
    // Billing Account. Projects can be linked to *at most* one Billing Account.
    ProjectIds []string

    // on_demand_billing_method configures the on-demand billing method for the
    // billing account. This billing method is expected to be charged with
    // on-demand cost that aren't covered by a contract or through a positive
    // credit balance.
    OnDemandBillingMethod *OnDemandBillingMethod

    // contract_billing_method configures a contract that can be attached to a
    // billing account to cover charges accrued.
```

```

ContractBillingMethod *ContractBillingMethod

// country is the billing account's country which is used for selecting
// country-specific prices and for invoicing.
Country          Country

// InternalOnDemand is a boolean that is true for a Company employee which
// replicates most onDemand payment features but prevents usage being
// charged
InternalOnDemand bool

// InternalContract is a boolean that is true for a Company employee which
// replicates contract functionality and prevents usage being charged
InternalContract bool
}

```

The `BillingAccountResponse` contains details of the OnDemand and Contract billing methods for display in the Cloud Portal UI. A new credit card type could be added to the brand enum or we could just utilize the existing UNKNOWN type. Default credit card values (e.g., 0/0 for the expiration date) could be returned in the response for now. See the UI section below to see how it would look utilizing the UNKNOWN type (which just means no credit card brand logo is displayed) without additional front-end code changes. An internal company brand could potentially be added to more clearly indicate the internal payment method in the future. This will require input from design and the front-end engineering team.

```

enum Brand {
    UNKNOWN = 0;
    AMERICAN_EXPRESS = 1;
    DINERS_CLUB = 2;
    DISCOVER = 3;
    JCB = 4;
    MASTERCARD = 5;
    UNION_PAY = 6;
    VISA = 7;
}

```

In the status quo, `AllowBillableRequest`, which is used by `RegisterResource`, checks the billing opt out flag via `ResourceBillingOptOut`. When deciding what package options to display in the UI `ListAvailablePackages` checks `isPackageAvailable` looking for onDemand or Contract billing methods.

Both of these will need to be updated to look at the internal field on the OnDemand and Contract billing methods as another way to grant access to the packages.

UI - Internal Contract

A button to open a modal for setting internal payment methods will be added to the billing account page of the Admin UI. Form validations will be used to ensure that only one or none of the internal payment method checkboxes are checked. Unchecking both boxes will result in the customer being removed from all internal payment methods and returned to an ACTIVE status where they can accrue charges.

Migration of Existing Internal Users

There are currently categories of two existing internal users: those who have added personal credit cards and are in the `billingOptOut` flag, and those who have been added to contracts.

For the users on contracts, we will move them to `internal_contract true`.

For the users on the feature flag, we will move them to `internal_onDemand true`. Then we will remove the code references to the feature flag, and eventually achieve and delete the feature flag itself.