



## Terraform New Worlds on MongoDB Atlas

Melissa Plunkett, Sr. Product Manager



themantissa

# Welcome!



- Sr. Product Manager in Cloud, focus Automation
- Formally a Solution Architect
- Before that an ops gal – Started with SGI IRIX and Red Hat pre RHEL.
- Tech Mantra: Learn it, Master it (enough). Script it. Rest. Repeat.
- And a Trekkie so ...



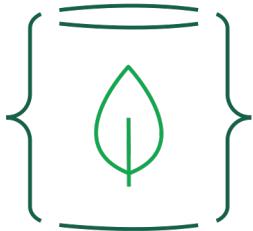


# Welcome Cadets!

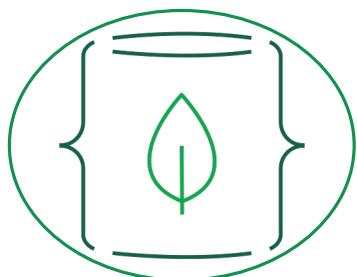
## Starfleet Academy for Automation



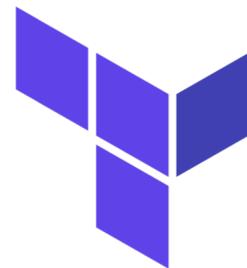
# Today's Training Agenda



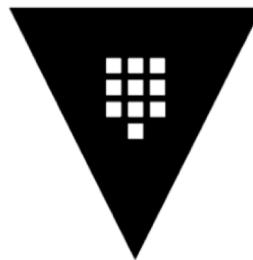
**MongoDB  
Atlas**



**MongoDB  
Atlas  
API**



**Terraform**



**Vault**

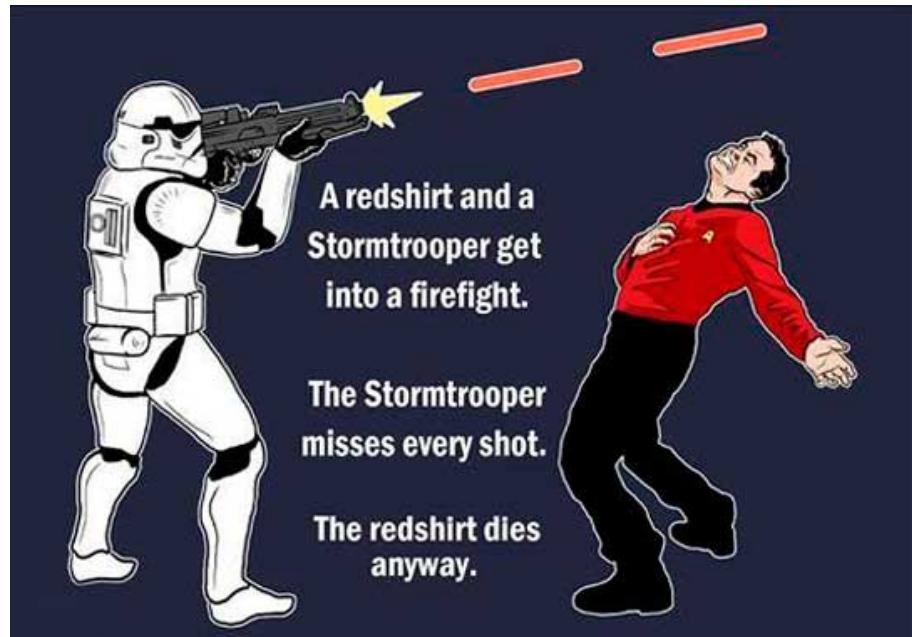


**Future  
Mission  
Plan**

# Cadets, Have You Used?



- MongoDB Atlas?
- MongoDB Atlas API?
- Terraform?
- Vault?
- Bonus!!! Star Trek Fan?  
Star Wars Fan? Both?



# MongoDB Atlas



Starfleet is focused on building applications that allow us to explore strange new worlds & seek out new civilizations

- MongoDB Atlas was selected earlier this year
  - Focus on apps not ops
  - Can quickly spin up on multiple cloud providers in multiple planetary\* regions!
  - Easy to use UI

# MongoDB Atlas UI Demo

# MongoDB Atlas – Exploring the API

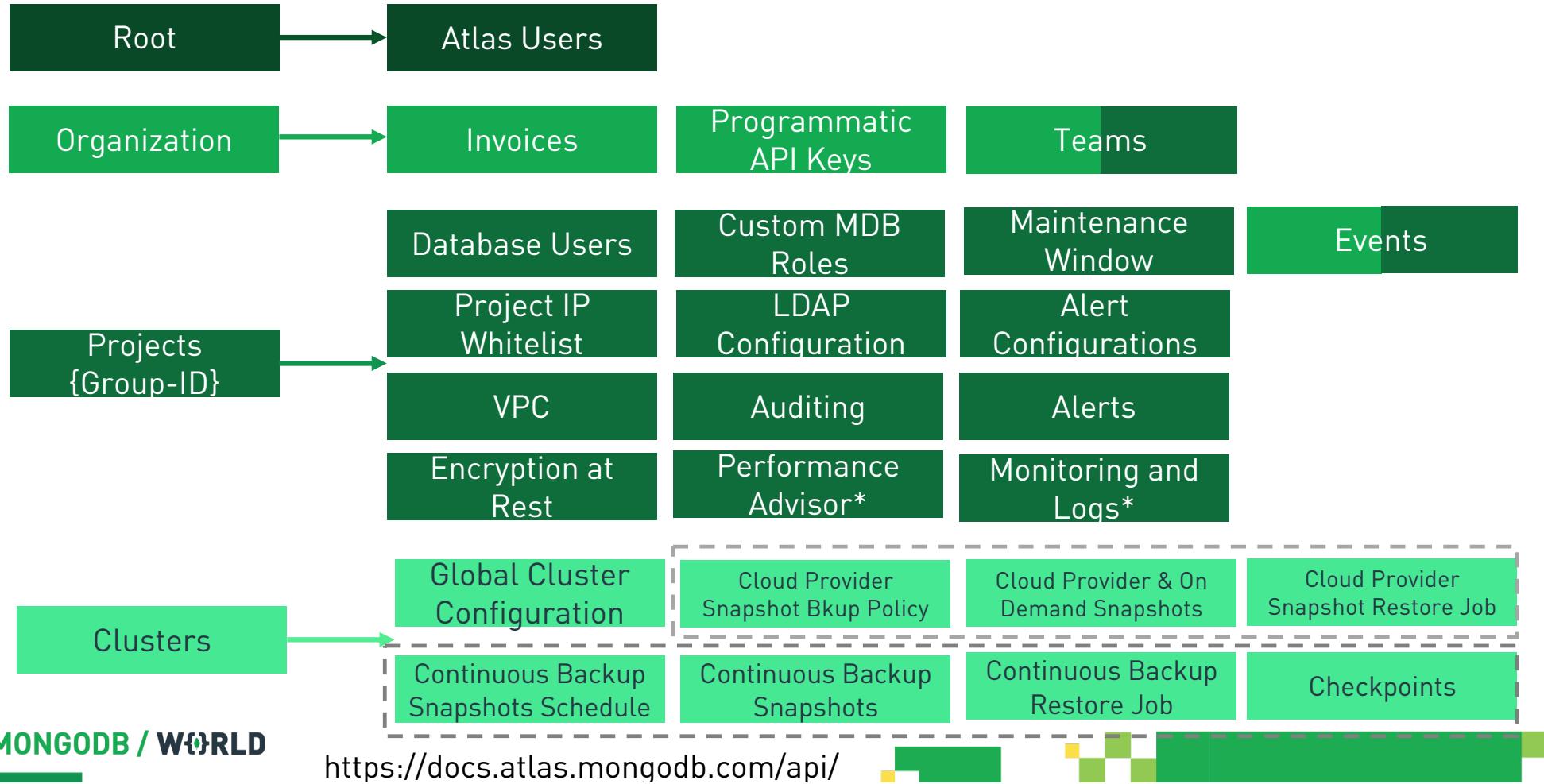


MongoDB Atlas API means UI isn't required to do much of what we need to do repeatedly to support developers

- REST
- API key w/ Digest Auth over HTTPS
- API keys has RBAC - capabilities match the permissions assigned to the Key
- API key whitelist

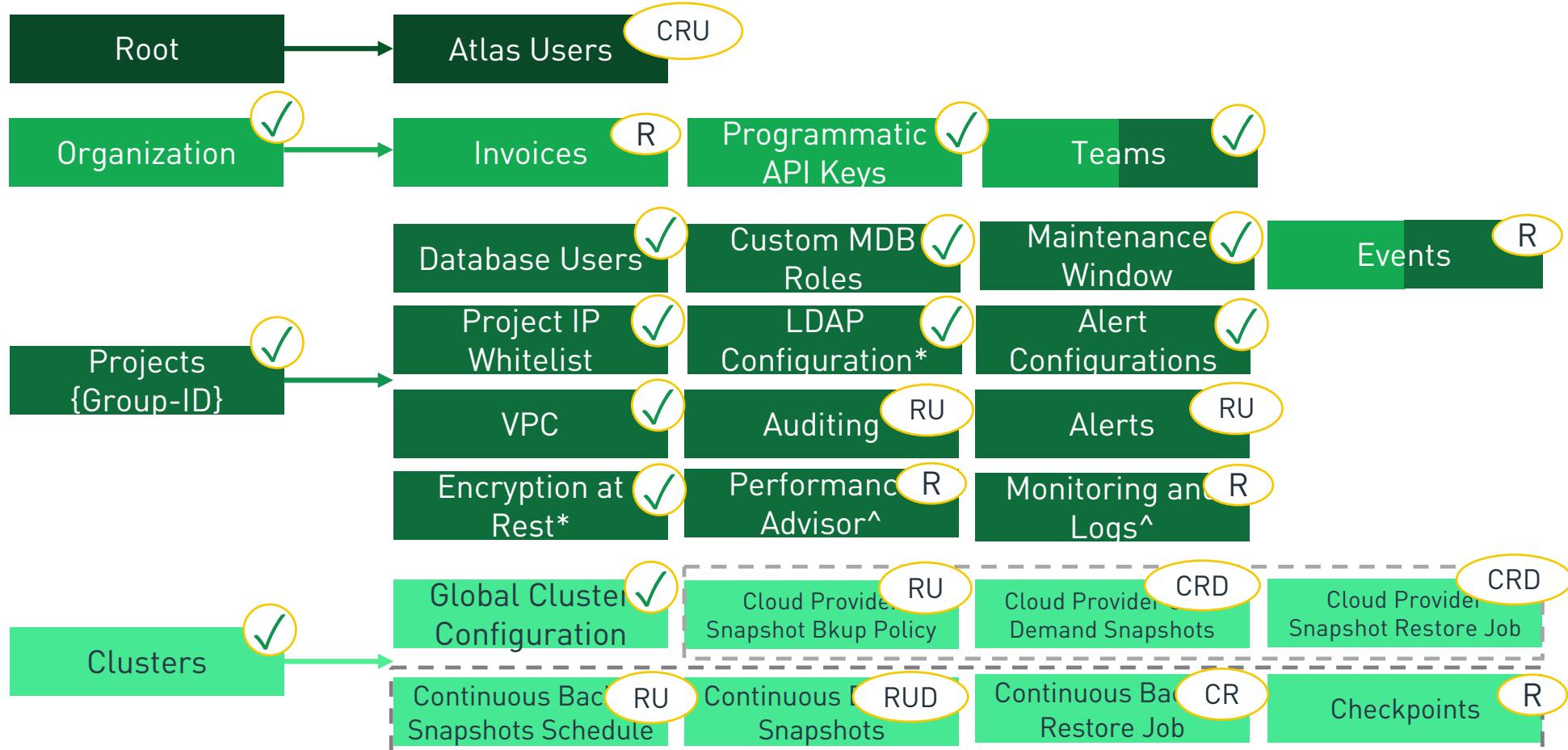


# MongoDB Atlas – Exploring the API - Resources





# MongoDB Atlas -API Resources Reference



# MongoDB Atlas – Exploring the API



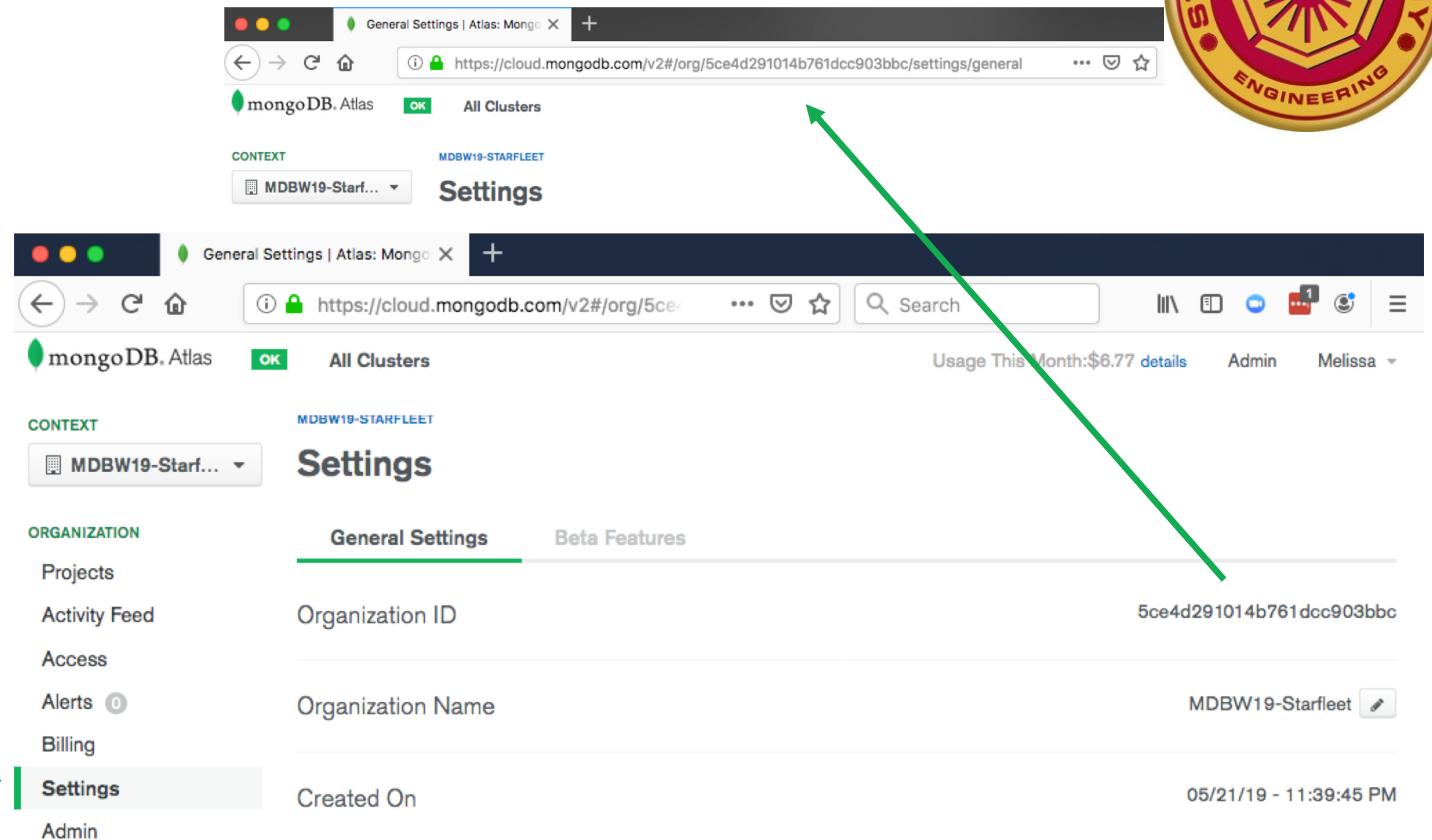
Two key things to know the location of before you get started ...

- Project ID
- Organization ID

# MongoDB Atlas – Exploring the API



Organization ID



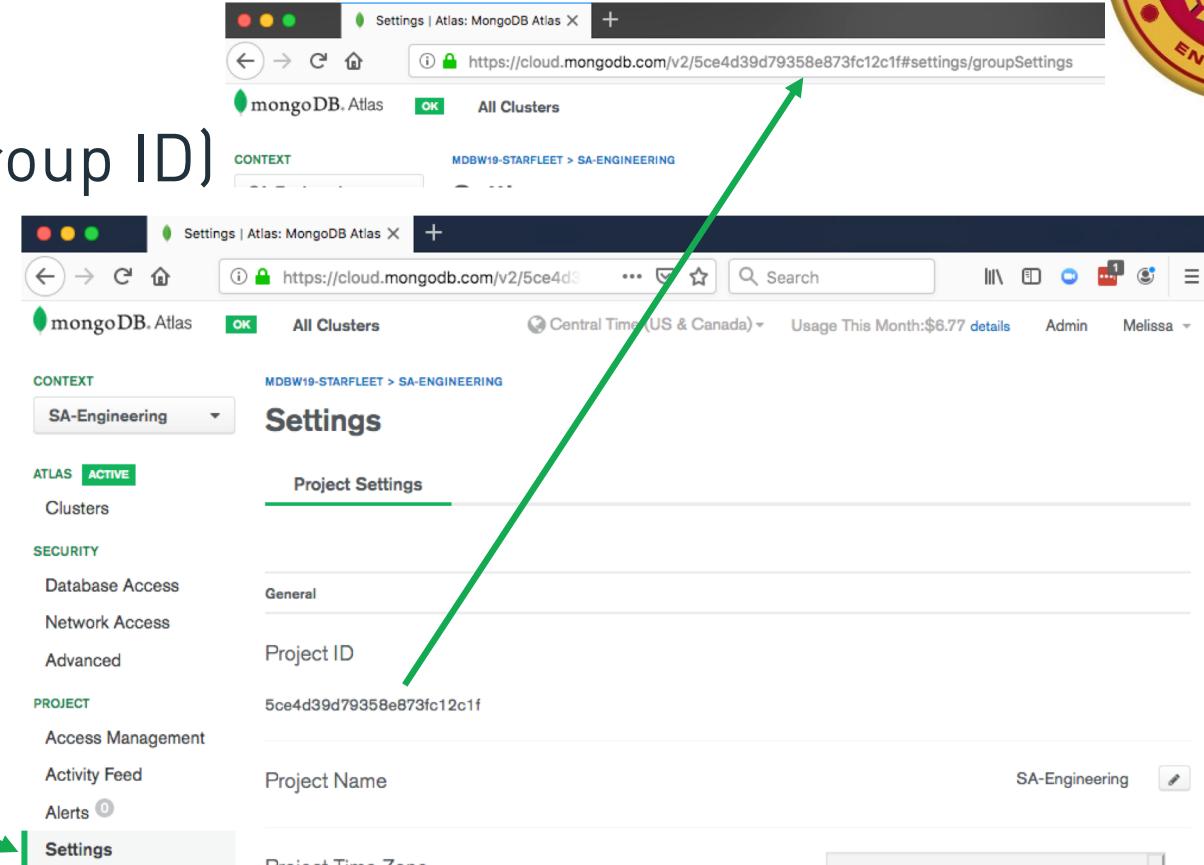
A screenshot of the MongoDB Atlas Settings page for the cluster "MDBW19-STARFLEET". The URL in the browser is <https://cloud.mongodb.com/v2#/org/5ce4d291014b761dcc903bbc/settings/general>. The page shows the "General Settings" tab selected under the "Organization" section. The "Organization ID" field is highlighted with a green arrow pointing from the left. The value "5ce4d291014b761dcc903bbc" is displayed next to it. Other fields shown include "Organization Name" (MDBW19-Starfleet) and "Created On" (05/21/19 - 11:39:45 PM). The left sidebar shows navigation options like Projects, Activity Feed, Access, Alerts, Billing, and Settings, with "Settings" currently selected.

Setting	Value
Organization ID	5ce4d291014b761dcc903bbc
Organization Name	MDBW19-Starfleet
Created On	05/21/19 - 11:39:45 PM

# MongoDB Atlas – Exploring the API



Project ID (aka Group ID)



A screenshot of the MongoDB Atlas Settings page. The URL in the browser is <https://cloud.mongodb.com/v2/5ce4d39d79358e873fc12c1f#settings/groupSettings>. The page shows the context "MDBW19-STARFLEET > SA-ENGINEERING". On the left, there's a sidebar with "ATLAS ACTIVE" selected, followed by "Clusters", "SECURITY", "Database Access", "Network Access", "Advanced", "PROJECT", "Access Management", "Activity Feed", "Alerts 0", and "Settings" which is highlighted with a green arrow. The main content area is titled "Settings" and shows "Project Settings" selected. Under "General", the "Project ID" is listed as "5ce4d39d79358e873fc12c1f". Below that, "Project Name" is set to "SA-Engineering". The "Project Time Zone" dropdown says "Please set your time zone". There are also decorative green bars at the bottom.

# MongoDB Atlas – Exploring the API



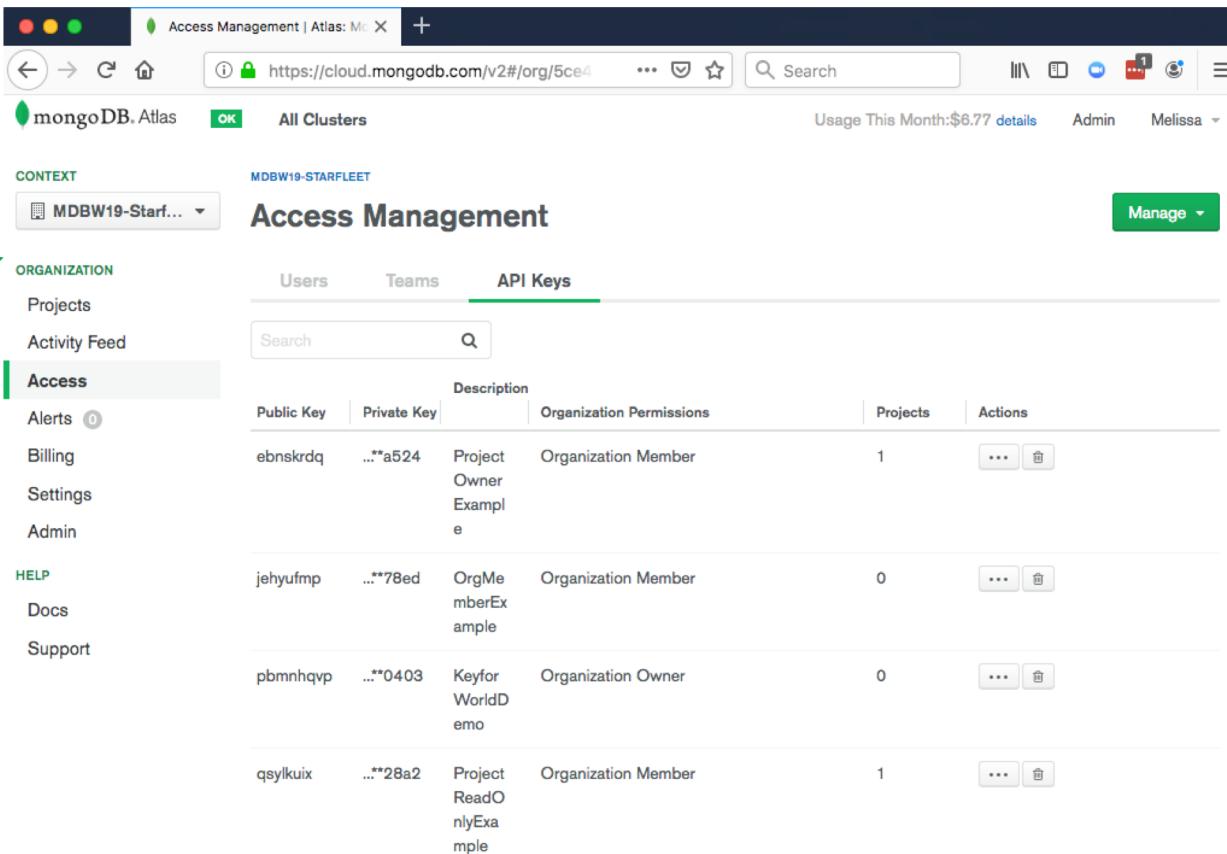
Create a Programmatic API Key either at the Organization or Project level:

- Org level: all the available permissions are for the org level.
- Project level: all the available permissions are for the project level
- BUT all project keys can be seen in the org level as an Org Member.

# MongoDB Atlas – Exploring the API



@ Org  
level go to  
Access



The screenshot shows the MongoDB Atlas Access Management interface. The left sidebar has a "CONTEXT" dropdown set to "MDBW19-STARFLEET". Under "ORGANIZATION", the "Access" item is highlighted with a green arrow pointing from the text above. Other items in the sidebar include "Projects", "Activity Feed", "Alerts (0)", "Billing", "Settings", "Admin", "HELP", "Docs", and "Support". The main area is titled "Access Management" and shows the "API Keys" tab selected. It lists four API keys:

Public Key	Private Key	Description	Organization Permissions	Projects	Actions
ebnskrdq	...**a524	Project Owner Example	Organization Member	1	[...] [Edit]
jehyufmp	...**78ed	OrgMemberExample	Organization Member	0	[...] [Edit]
pbnmhqvp	...**0403	KeyforWorldDemo	Organization Owner	0	[...] [Edit]
qsylkuix	...**28a2	Project ReadOnlyExample	Organization Member	1	[...] [Edit]

# MongoDB Atlas – Exploring the API



@ Project level go to Access Management



Screenshot of the MongoDB Atlas Access Management interface for the "SA-Engineering" project.

The sidebar shows the following navigation options:

- CONTEXT: SA-Engineering
- PROJECT:
  - Access Management (highlighted)
  - Activity Feed
  - Alerts (0)
  - Settings
- SERVICES:
  - Charts
  - Stitch
- HELP:
  - Docs
  - Support

The main content area displays the "Access Management" page for the "SA-Engineering" project. The "API Keys" tab is selected. The table lists two API keys:

Public Key	Private Key	Description	Project Permissions	Projects	Actions
ebnskrdq	...**a524	ProjectOwnerExample	Project Owner	1	[...] [Delete]
qsylkuix	...**28a2	ProjectReadOnlyExample	Project Read Only	1	[...] [Delete]

System Status: All Good Last Login: 99.88.244.81 Version: 478328d22f@v20190528  
Atlas Plan: NDS Effective Plan: NDS Plan Start Date: 2019-05-22T04:39:45Z Central URL: https://cloud.mongodb.com  
Organization Name: MDBW19-Starfleet  
©2019 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

# Create a Cluster:

```
--request POST  
"https://cloud.mongodb.com/api/atlas/v1.0/groups/5ce4d39d79358  
e873fc12c1f/clusters?pretty=true" \  
--data '  
{  
  "name" : "MDBW19-Class-Cluster2-API",  
  "diskSizeGB" : 100,  
  "autoScaling" : {  
    "diskGBEnabled" : false  
  }, "clusterType": "REPLICASET",  
  "providerBackupEnabled" : true,
```

# Create a Cluster:

```
"providerSettings" : {  
    "providerName" : "AWS",  
    "diskIOPS": 340,  
    "instanceSizeName" : "M30",  
    "volumeType" : "PROVISIONED" },
```

## Create a Cluster:

# Create a Cluster:

```
"US_EAST_1" : {  
    "analyticsNodes" : 0,  
    "readOnlyNodes" : 1,  
    "electableNodes" : 2,  
    "priority" : 6 }  
}  
}  
}  
}
```

# Create a MongoDB Database User in the Project:

```
-X POST  
"https://cloud.mongodb.com/api/atlas/v1.0/groups/5ce4d39d79358  
e873fc12c1f/databaseUsers?pretty=true" \  
--data '  
{ "databaseName" : "admin",  
  "roles" : [ {  
      "databaseName" : "admin",  
      "roleName" : "readWriteAnyDatabase"  
    } ],  
  "username" : "spock2",  
  "password" : "badpass123" } '
```

# Add an IP to the Project IP Whitelist:

```
--request POST
"https://cloud.mongodb.com/api/atlas/v1.0/groups/5ce4d39d79358
e873fc12c1f/whitelist?pretty=true" \
--data '[
{
    "ipAddress" : "192.0.0.15",
    "comment" : "IP address added via API"
}]'
```

# Create a Quick Cluster via API

# MongoDB Atlas – Exploring the API



API is great but...

- Requires us to really know the API: e.g. updating via PATCH, or delete via DELETE, etc.
- Going to possibly take a lot of time and ~~money~~.
- What if we could just describe what we want and change that when we need to modify it?

**Emergency Transmission . . .**

**Top Secret**

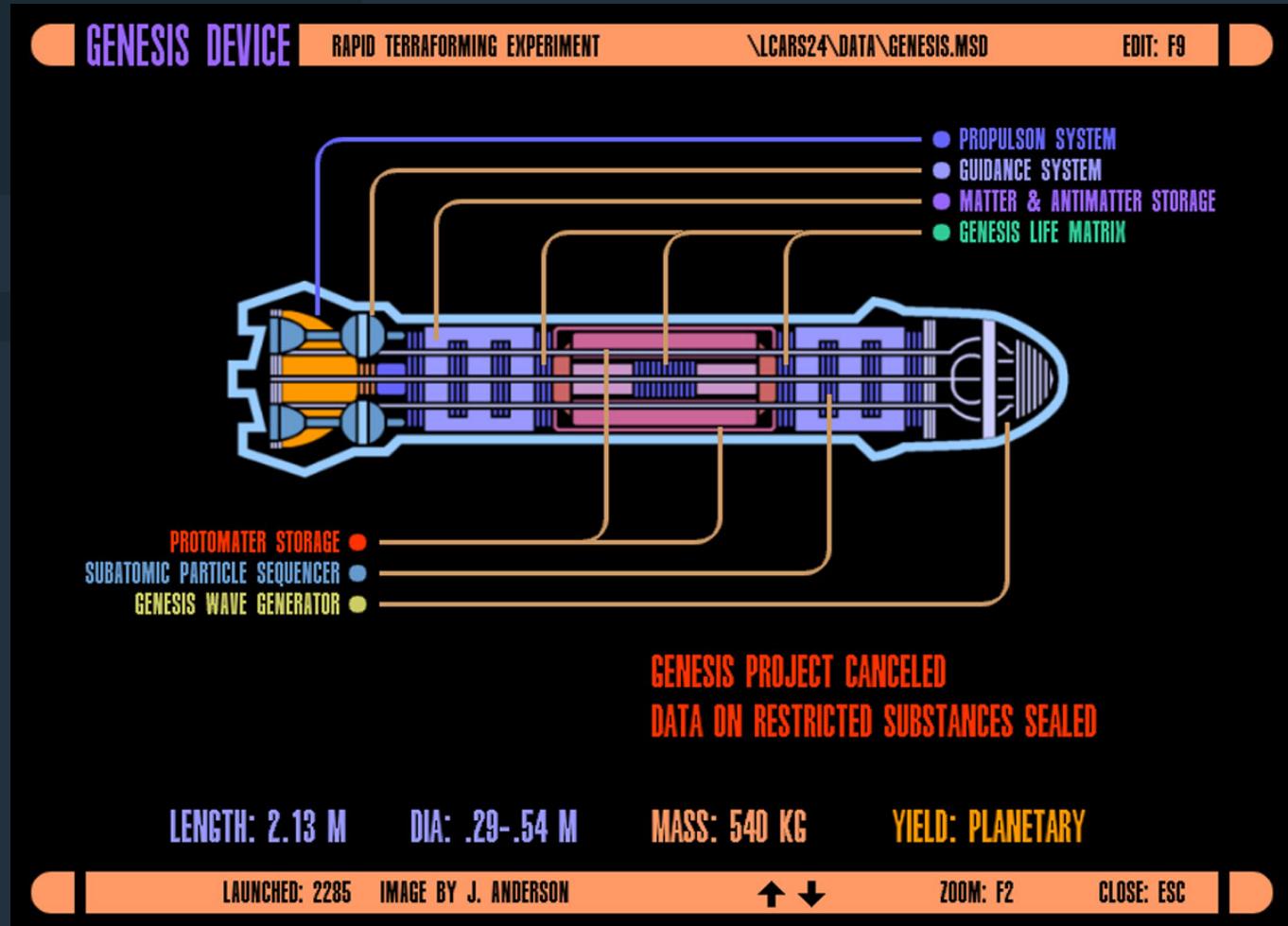
**From: Starfleet Command**

**To: Starfleet Engineering**

**MongoDB Cluster needed STAT. Project  
Genesis is at risk. Deploy at once!**

**Khan seen in sector. Red Alert recommended.**

# Emergency Transmission . . .



# HashiCorp's Terraform



Terraform is an open source, declarative Infrastructure as Code (IaC) tool.

- Describe desired infrastructure in a configuration file
- Terraform takes that file and creates a plan to implement it
- One can review the plan, and if all good, apply to create
- We can create, change and destroy infrastructure easily!

# HashiCorp's Terraform



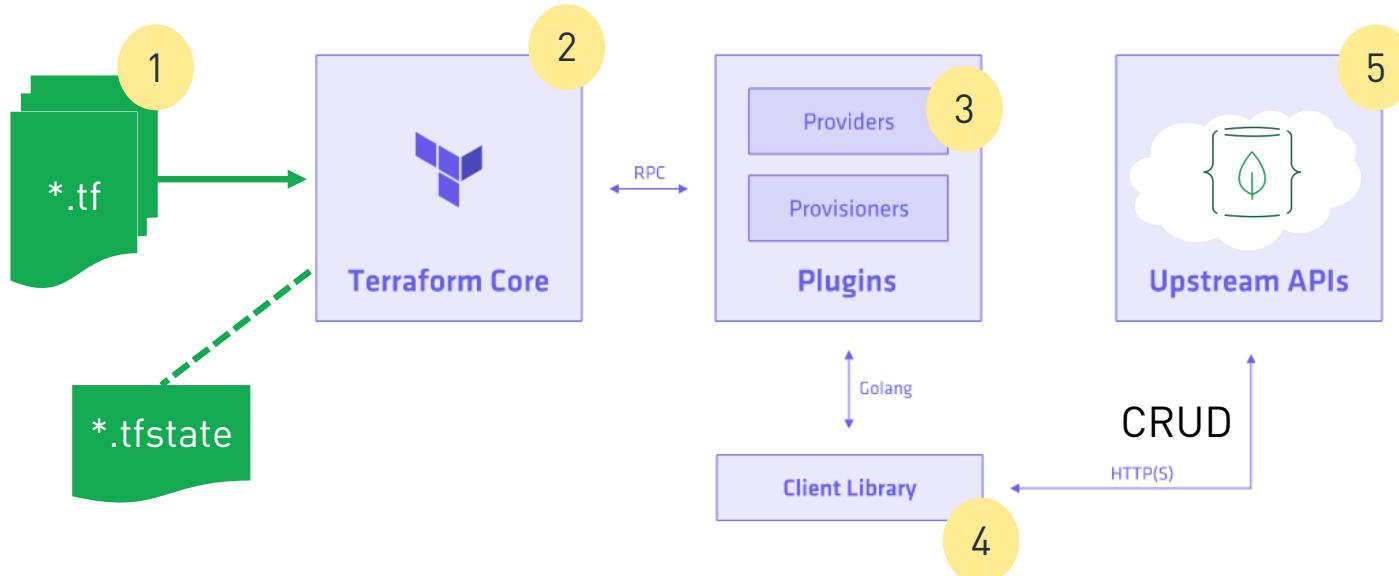
We can change and destroy infrastructure easily???

- Terraform keeps a state file of what it has done
- Change your configuration it only implements the diff
- Can import a resource and then apply changes, e.g. import in a cluster and then set it to a bigger size.
- Terraform destroy really destroys!



# HashiCorp's Terraform

Terraform is a plugin based system:



The Provider understands both Terraform and the Upstream API

Terraform Core doesn't "know" any infrastructure, that's the Plugins purpose.

You simply need to define what you want in the `.tf` file, do not need to know the API deeply



# Main.tf - Variables

```
variable "mongodb_atlas_api_pub_key"  
        { default = "PUB-API-KEY"}  
  
variable "mongodb_atlas_api_pri_key"  
        { default = "PRI-API-KEY" }  
  
variable "database_username" { default = "spock3" }  
variable "database_user_password" { default = "blahblah1236" }  
variable "mongodb_atlas_whitelistip" { default = "192.155.1.1" }
```

variable "mongodb\_atlas\_api\_pub\_key"  
 { default = "PUB-API-KEY" }

variable "mongodb\_atlas\_api\_pri\_key"  
 { default = "PRI-API-KEY" }

Keys and passwords !?

Not a great practice and angers Admiral Kirk as Khan could steal these ☹  
But we'll address this soon.



KHAAN

# Main.tf – Configure the Provider

```
# Configure the MongoDB Atlas Provider
provider "mongodbatlas" {
    username = "${var.mongodb_atlas_api_pub_key}"
    api_key = "${var.mongodb_atlas_api_pri_key}"
}
```

# Main.tf - Cluster

```
resource "mongodbatlas_cluster" "cluster" {  
    name          = "MDBW19-Class-Cluster3-TF"  
    group         = "${var.mongodb_atlas_project_id}"  
    mongodb_major_version = "4.0"  
    provider_name      = "AWS"  
    region           = ""  
    size             = "M30"  
    disk_size_gb     = 100  
    backup           = false  
    provider_backup   = true  
    disk_gb_enabled  = false  
    replication_factor = 0
```

**Missing:** A few issues  
-Disk IO,   
-Anything related to  
replicationSpecs - this uses the  
deprecated replicationSpec  
option.  
-numShards (but it has a default!)

# Main.tf – Cluster Continued

```
replication_spec {  
    region          = "US_WEST_1"  
    priority        = 7  
    read_only_nodes = 0  
    analytics_nodes = 1  
    electable_nodes = 3  
}
```

```
replication_spec {  
    region          = "US_EAST_1"  
    priority        = 6  
    read_only_nodes = 1  
    analytics_nodes = 0  
    electable_nodes = 2  
}  
}
```

# Main.tf – Database User

```
# Create a Database User

resource "mongodbatlas_database_user" "test" {

    username = "${var.database_username}"
    password = "${var.database_user_password}"
    database = "admin"
    group     = "${var.mongodb_atlas_project_id}"

    roles {
        name = "readWriteAnyDatabase"
        database = "admin"
    }
}
```

# Main.tf – IP Whitelist

```
# Create an IP Whitelist

resource "mongodbatlas_ip_whitelist" "test" {
    group      = "${var.mongodb_atlas_project_id}"
    ip_address = "${var.mongodb_atlas_whitelistip}"
    comment     = "Added with Terraform"
}
```

# HashiCorp's Terraform



Let's configure a cluster, MongoDB user and Project IP whitelist with Terraform

- Providers can be verified/official (both Hashicorp & Community created) OR community created but not verified
- MongoDB Atlas Community Provider created by Akshay Karle: <https://github.com/akshaykarle/terraform-provider-mongodbatlas/> (Thank you Akshay!)

# Create a Quick Cluster via Terraform

# Cadets Save Day and Defeat Khan!





# Security of the Mission

We have to ensure we handle data securely. Security layers we have utilized thus far:

- API communication over HTTPS
- RBAC to limit what the API key can do
- IP Whitelist

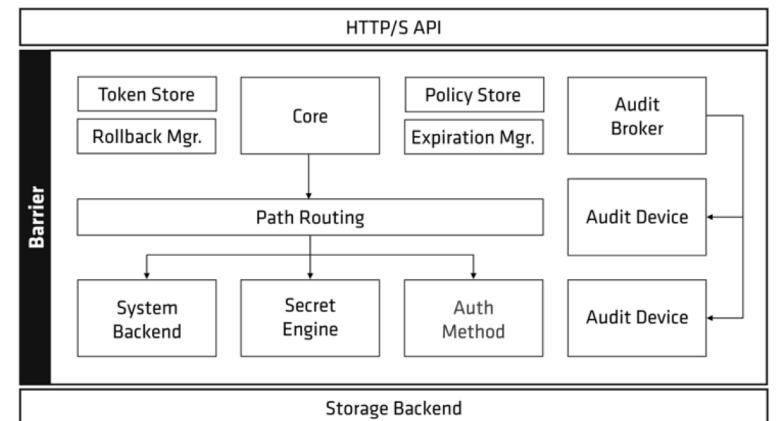
But the more secure layers the better so what else can we do?



# HashiCorp's Vault

Vault is another HashiCorp product designed for security in the Cloud:

- Manage secrets & protect sensitive data
- Focus here - managing secrets, especially dynamic secrets
- Secret Engine extends Vault to systems, e.g. physical like a HSM or a service in the Cloud like Atlas.





# HashiCorp's Vault

- Database Secrets Engine: generate database credentials dynamically based on configured roles, including for MDB.
- Atlas creates database users differently (via API) so need a Secrets Engine for that.
- Ta da! There's a community Secrets Engine at <https://github.com/mealal/vault-atlas-plugin> written by the amazing Alexey Menshikov, a Consulting Engineers @MDB!



# HashiCorp's Vault

- Atlas community Secrets Engine examples:

```
$ vault write database/roles/readonly db_name=atlas  
creation_statements='{ "db": "admin", "roles": [{ "role":  
"readAnyDatabase" }] }' default_ttl="1h" max_ttl="24h"  
  
$ vault write database/config/atlas plugin_name=atlas  
allowed_roles="readonly" apiID="public API key"  
apiKey="private API key" groupID="project id"  
  
$ vault read database/creds/readonly
```



# HashiCorp's Vault

- What about a Provider for Vault? Yes!  
<https://www.terraform.io/docs/providers/vault/index.html>
- Read the cautions on using the Provider for Vault and follow the best practices to be as secure as possible!



# Cadets Save Day and Defeat Khan!



# **Future Mission Plan**



# MongoDB + Hashicorp – Better together!

Later in 2019 MongoDB will have an official:

- MongoDB Atlas Terraform Provider
- MongoDB Atlas Secrets Engine for Vault



# Be the First in the Know!



- Go to <https://bit.ly/2WS4WMJ>
- Fill out the brief survey, 6 required q's, 3 not! < 2m
- You will be notified on the progress for which ever mission you are interested in (Vault/Terraform/both)



# Questions?

Where: Atlas Booth in the Partner Pavilion (A1)

When: 1:40 - 5:05 PM today!

or

[Melissa.Plunkett@mongodb.com](mailto:Melissa.Plunkett@mongodb.com)  
anytime!

 Github: themantissa



**YOU ARE IN REGENT**

**Please provide Session Feedback**

**1. MDB World App -> Menu -> “Rate a Session” ->  
Regent**

**OR**

- 1. Go to [slido.com](https://www.slido.com)**
- 2. Enter event code #MDBW19**
- 3. Click on Regent**

***Feedback poll will remain open for 10 minutes after the talk ends***

**Thank you!**



# MONGODB / WORLD