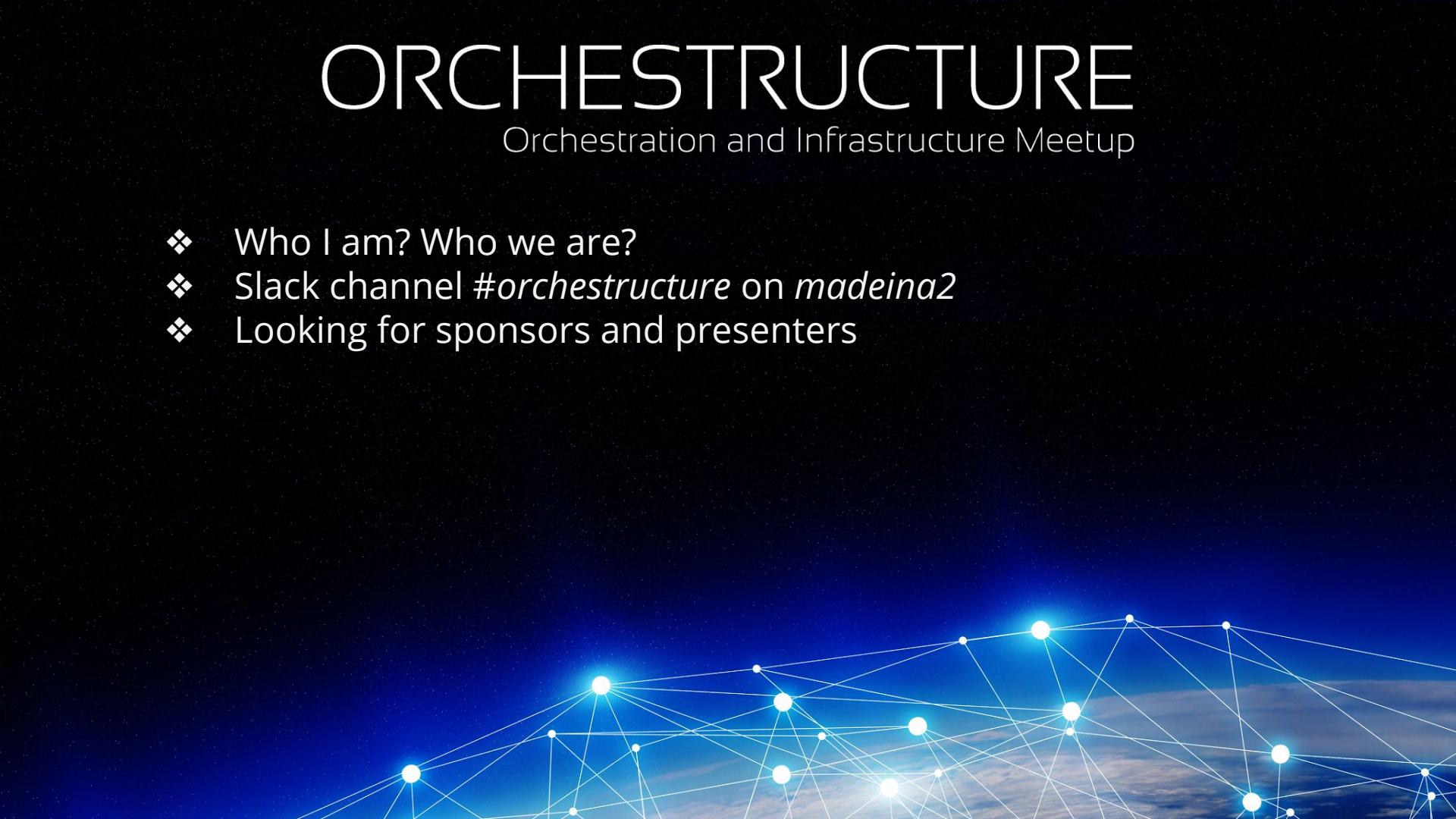
ORCHESTRUCTURE

Orchestration and Infrastructure Meetup

- Who I am? Who we are?
- Slack channel *#orchestructure* on *madeina2*
- Looking for sponsors and presenters

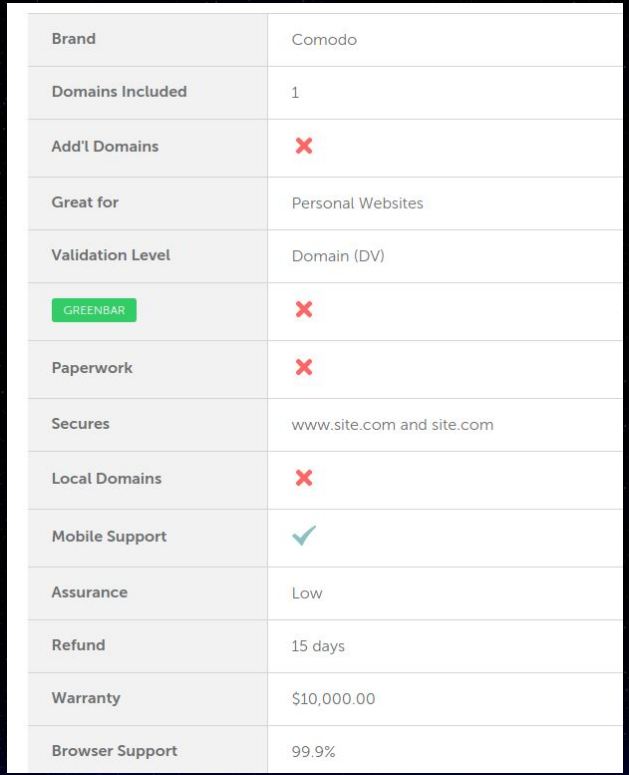


Dynamic Certificate Generation:
- Welcome to ACME

- Automated Free Certificates
- Modern Web Servers
- Dynamic Certificates
- Best Practices

Handling Certificates Before

- Purchase from CA like Comodo
- Go through painful verification process
- Manually provide to Web Server
- Entirely Manual Process
- But supported (warranty, refund)
- Still applicable based on situation

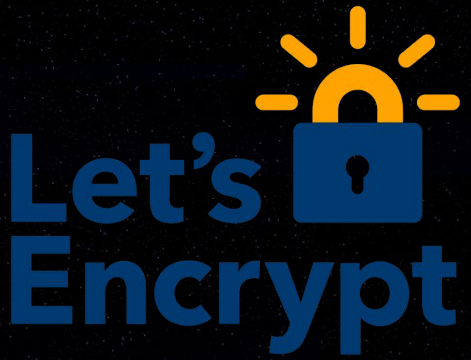


Brand	Comodo
Domains Included	1
Add'l Domains	✗
Great for	Personal Websites
Validation Level	Domain (DV)
GREENBAR	✗
Paperwork	✗
Secures	www.site.com and site.com
Local Domains	✗
Mobile Support	✓
Assurance	Low
Refund	15 days
Warranty	$10,000.00
Browser Support	99.9%

Making it Easier

- Cloudflare and others serve as Reverse Proxy
- Lots of guides based on Webserver
- Super cheap; StartSSL, etc..

Enter LetsEncrypt



- Automatically Generated
- Completely Free
- API Driven
- Open CA via ISRG (https://letsencrypt.org/isrg/)
- Multiple Verification Methods (HTTP, DNS, TLS)
- Support for your WebServer™

** User Friendly!
https://letsencrypt.org/how-it-works/

The Power of ACME

- Automated Certificate Management Environment
- Standardized method of managing Certificates
- Its own IETF Working Group
- Currently being drafted (in-progress)
- https://github.com/ietf-wg-acme/acme

This document describes a protocol that a certification authority (CA) and an applicant can use to automate the process of verification and certificate issuance. The protocol also provides facilities for other certificate management functions, such as certificate revocation.

Modern Web Servers

```
your.public.com

proxy / localhost:9000 {
    transparent
}
```

- Caddy and Traefik
- Simplistic or Dynamic Configuration
- Container/Cloud-ready (written in Go)
- Feature Rich - Plugin Architecture
- Automatically handle LE Challenges
- New Reverse proxy services like Fly.io

You got served by **Caddy**

Certificates in a Cluster

- *Define*, *Challenge*, *Store*
- Individual Service with one purpose
  - Kube-lego, rancher-letsencrypt
- Definition either passed at start or Ingress Rule
- Challenge either HTTP or DNS
- Store, either local fs or secrets

Load Balancer Handling

- Ingress Rule definition
- Pointer to Certificate to utilize for said domain (secret)
- Host based matching on domain (req. SNI)

```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: echoserver
  namespace: echoserver
  annotations:
    kubernetes.io/tls-acme: "true"
    kubernetes.io/ingress.class: "nginx"
spec:
  tls:
  - hosts:
    - echo.example.com
    secretName: echoserver-tls
  rules:
  - host: echo.example.com
    http:
      paths:
      - path: /
        backend:
          serviceName: echoserver
          servicePort: 80
```

Best Practices in Production

- https://cipherli.st/
- OWASP Cheat Sheet
- SSLLabs Best Practices
- SSLLabs Test
- Things like HSTS, session tickets (timeouts), OCSP

nginx

```
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_ciphers "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+ED
H";
ssl_ecdh_curve secp384r1; # Requires nginx >= 1.1.0
ssl_session_cache shared:SSL:10m;
ssl_session_tickets off; # Requires nginx >= 1.5.9
ssl_stapling on; # Requires nginx >= 1.3.7
ssl_stapling_verify on; # Requires nginx => 1.3.7
resolver $DNS-IP-1 $DNS-IP-2 valid=300s;
resolver_timeout 5s;
add_header Strict-Transport-Security "max-age=63072000; incl
udeSubDomains; preload";
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;
```

ORCHESTRUCTURE

Orchestration and Infrastructure Meetup

- Thanks for Tuning in!
- Keep an eye on Meetups for next meeting details
- Jump in Slack (#orchestructure)