



DRUPAL DEVELOPER DAYS
LISBON 2018

It Works on My Machine (!)

Christopher Torgalson

Diamond Sponsor

AcQUia®



Platinum Sponsors



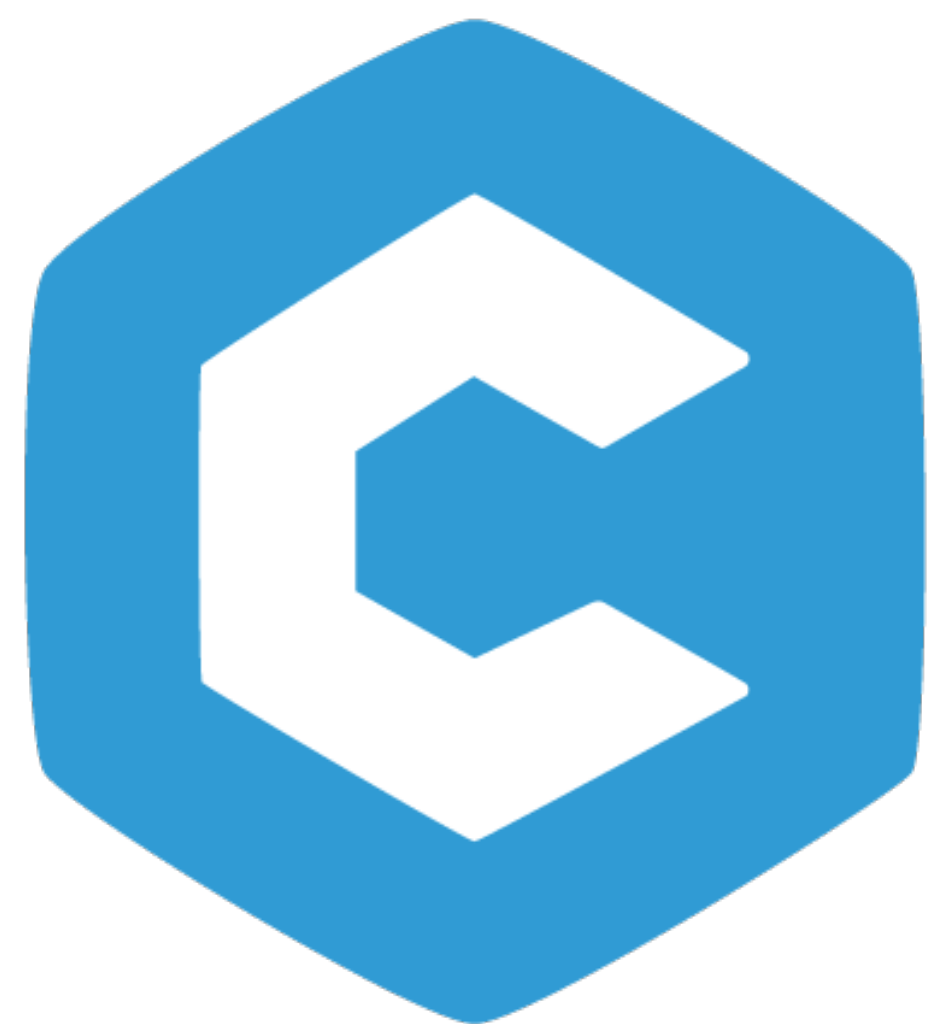
Gold Sponsors



Christopher Torgalsen

@bedlamhotel →





CHROMATIC

chromatichq.com



Glossary



Dev

Any local development instance



Prod

Any remote server—might not *actually* be Production



Provision

Building out and configuring a server



Deploy

Building an application on a provisioned server

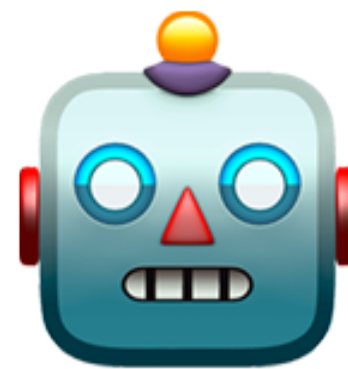


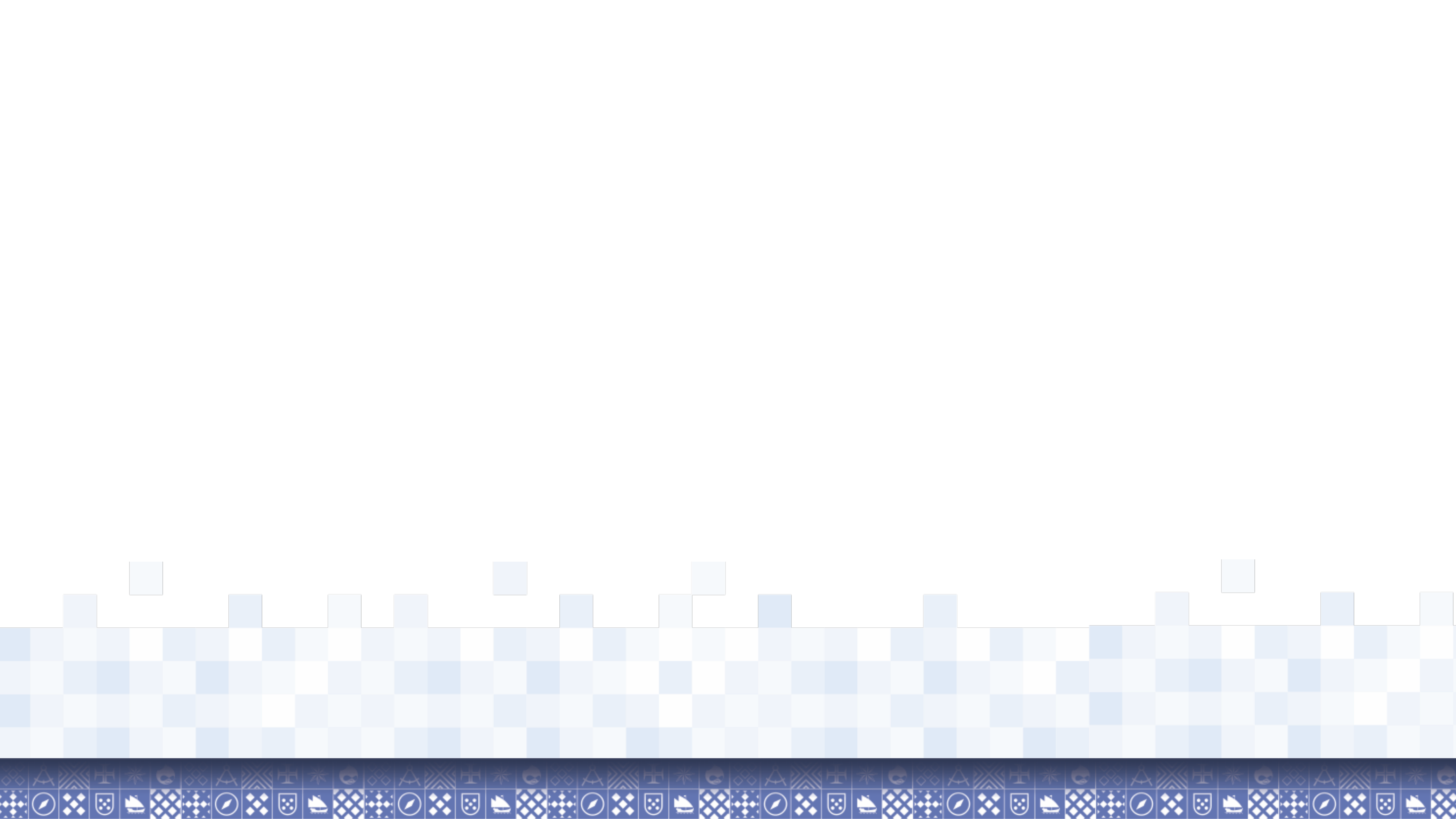
Automated deployment



Automated deployment

Manual deployment by robot!





Dev is Hard



Prod is Not



Prod is Not



So...



Do



Do Dev



Do Dev Like

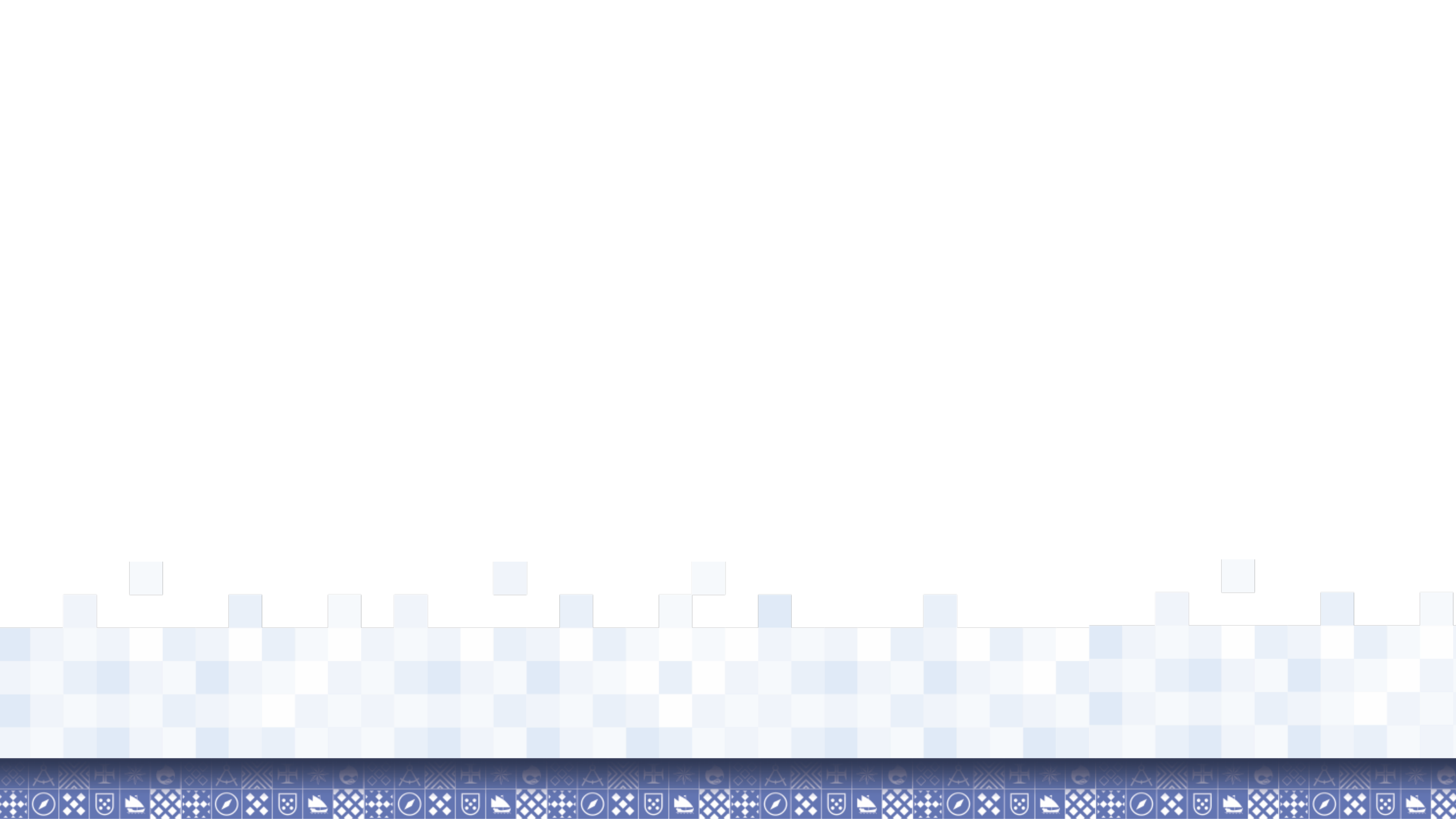


Do Dev Like Prod



(spoiler)





> `ansible-playbook deployment.yml`



Requirements



A solution must be able to:



A solution must be able to:

1. Configure servers and deploy applications



A solution must be able to:

1. Configure servers and deploy applications
2. Run tasks on the controller, dev, and prod



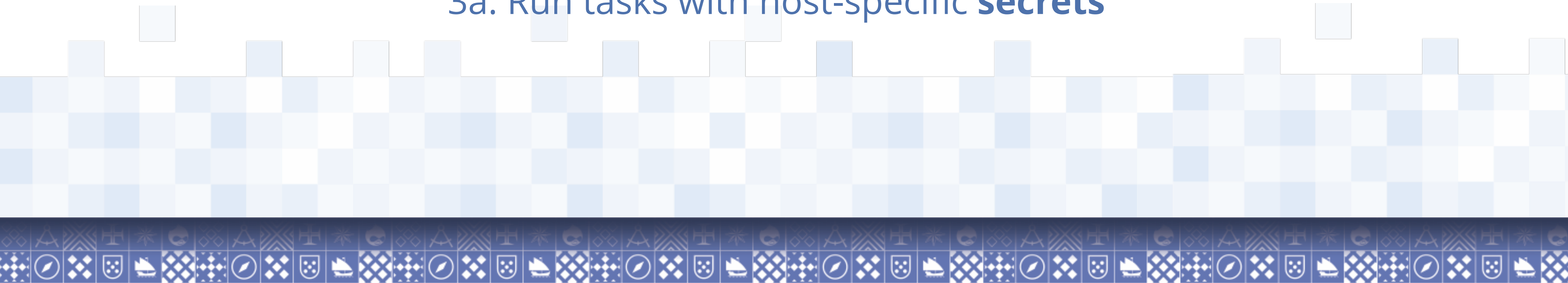
A solution must be able to:

1. Configure servers and deploy applications
2. Run tasks on the controller, dev, and prod
3. Run tasks with host-specific variables



A solution must be able to:

1. Configure servers and deploy applications
2. Run tasks on the controller, dev, and prod
3. Run tasks with host-specific variables
- 3a. Run tasks with host-specific **secrets**



Tooling



Vagrant



Vagrant

virtual machine(s)



Vagrant

virtual machine(s)

code-only configuration



Vagrant

virtual machine(s)

code-only configuration

Vagrantfile →



Vagrant

```
VAGRANTFILE_API_VERSION = "2"

Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  config.vm.box = "bento/ubuntu-16.04"
end
```



Ansible



Ansible

task runner



Ansible

task runner

“automation glue”



Ansible

task runner

“automation glue”

tasks, plays, and playbooks



Ansible

task runner

“automation glue”

tasks, plays, and playbooks

devops/ →



Tasks



Tasks

commands



Tasks

commands

task-specific modules



Tasks

commands

task-specific modules

devops/ →



Tasks

```
- name: Run Gulp build.  
  command: "npx gulp build"  
  args:  
    chdir: "../site/docroot"  
  delegate_to: localhost
```



Play(book)s



Play(book)s

collections of tasks or plays



Play(book)s

collections of tasks or plays

deployment.yml →



Play(book)s

```
- name: Deploy application.  
  hosts:  
    - deploy_dev  
    - deploy_prod  
  gather_facts: true  
  tags:  
    - deploy  
  
  tasks:  
    - name: Run application deployment tasks.  
      include_tasks: "tasks/deploy-app.yml"
```

Tags



Tags

filter *tasks* and *plays* to run on hosts



Tags

filter *tasks* and *plays* to run on hosts

--tags / -t



Tags

filter *tasks* and *plays* to run on hosts

--tags / -t

deployment.yml →



Tags

filter *tasks* and *plays* to run on hosts

--tags / -t

deployment.yml →

Vagrantfile →



> `ansible-playbook deployment.yml`

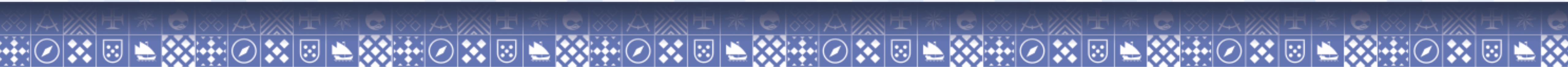


```
> ansible-playbook deployment.yml  
--tags=deploy
```



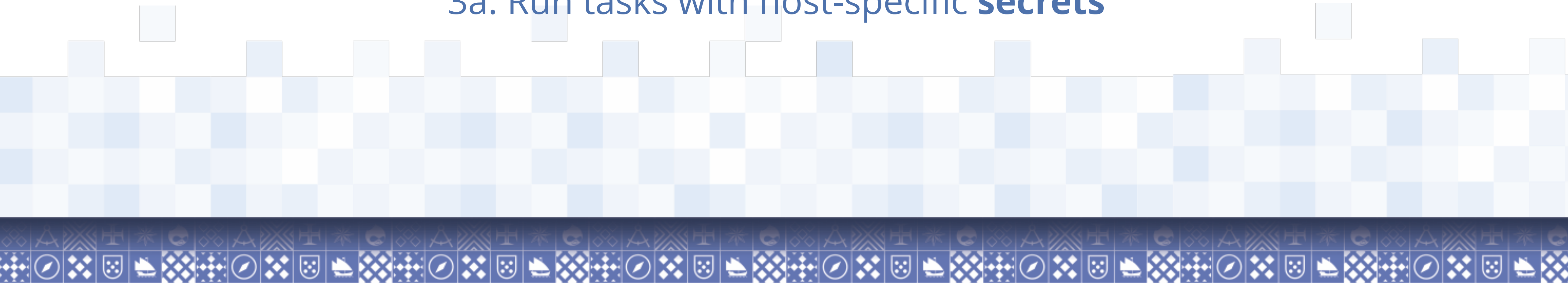
● ● ● ctorgalson@bh-Macmini7-1: ~/deploy-everywhere/devops

→ devops git:(master) x ansible-playbook deployment.yml -t deploy



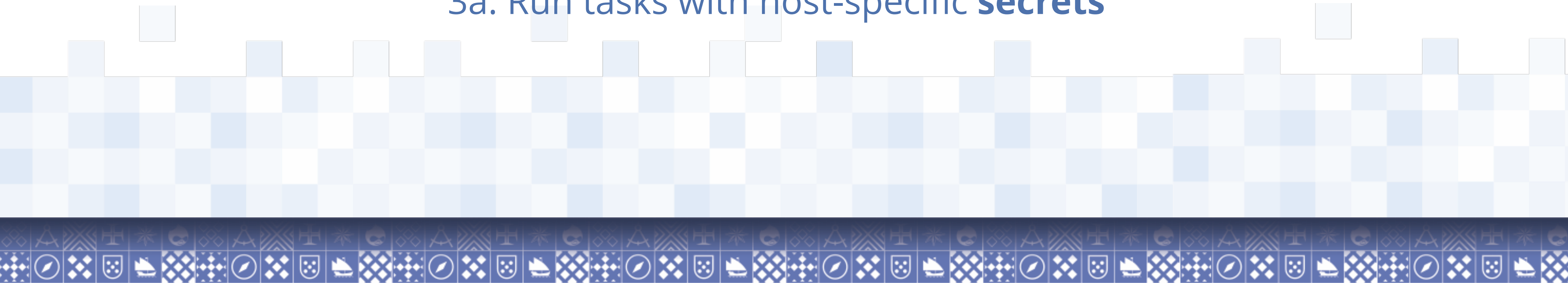
A solution must be able to:

1. Configure servers and deploy applications
2. Run tasks on the controller, dev, and prod
3. Run tasks with host-specific variables
- 3a. Run tasks with host-specific **secrets**



A solution must be able to:

- ~~1. Configure servers and deploy applications~~
- 2. Run tasks on the controller, dev, and prod
- 3. Run tasks with host-specific variables
- 3a. Run tasks with host-specific **secrets**



Groups



Groups

define *hosts* to run tasks on



Groups

define *hosts* to run tasks on

```
--limit / -l
```



Groups

define *hosts* to run tasks on

`--limit / -l`

hosts.yml →



Groups

```
all:
  children:
    deploy_dev:
      hosts:
        dev-01.deploy.local:
          ansible_host: 192.168.101.202
```

> `ansible-playbook deployment.yml`

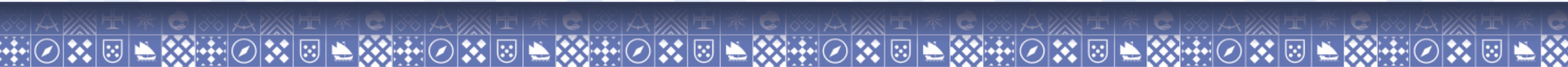


```
> ansible-playbook deployment.yml  
--limit=deploy_dev
```



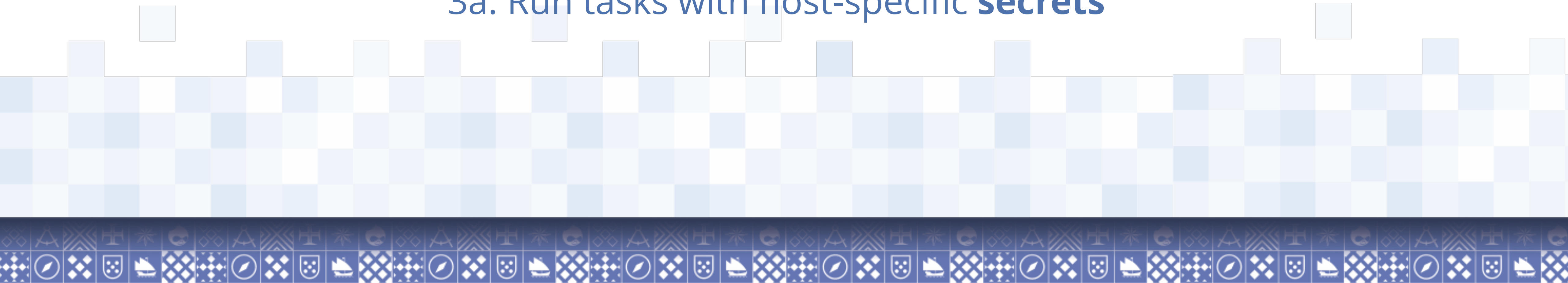
● ● ● ctorgalson@bh-Macmini7-1: ~/deploy-everywhere/devops

→ devops git:(master) x ansible-playbook deployment.yml -l deploy_dev



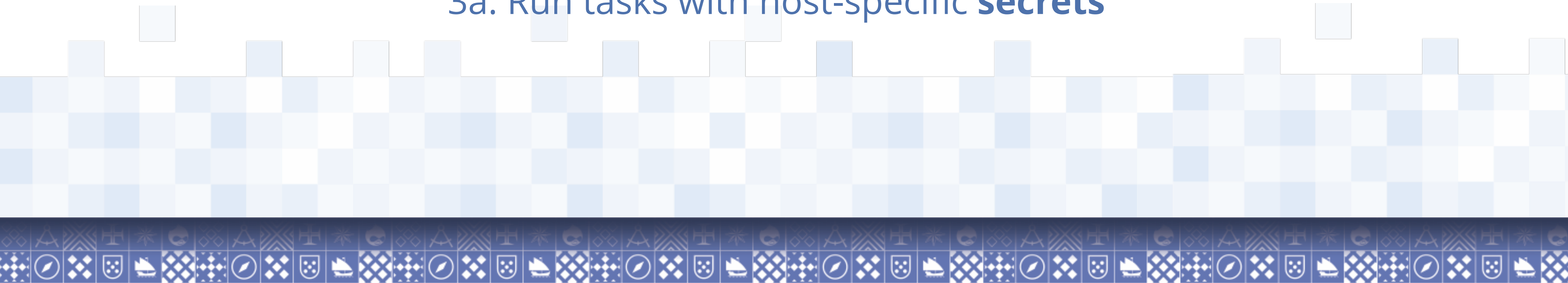
A solution must be able to:

- ~~1. Configure servers and deploy applications~~
- 2. Run tasks on the controller, dev, and prod
- 3. Run tasks with host-specific variables
- 3a. Run tasks with host-specific **secrets**



A solution must be able to:

- ~~1. Configure servers and deploy applications~~
- ~~2. Run tasks on the controller, dev, and prod~~
- 3. Run tasks with host-specific variables
- 3a. Run tasks with host-specific **secrets**



Group Variables



Group Variables

assign variables per host *group*



Group Variables

assign variables per host *group*

auto-loaded from files or directories (+`"all"`)



Group Variables

assign variables per host *group*

auto-loaded from files or directories (+*"all"*)

group_vars/ →



Group Variables

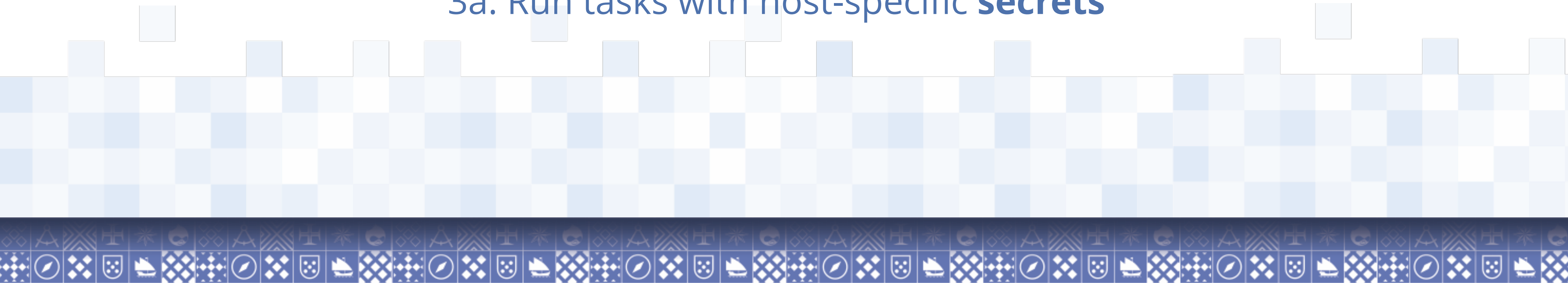
```
all:
  children:
    deploy_dev:
      hosts:
        dev-01.deploy.local:
          ansible_host: 192.168.101.202
```


Group Variables

```
all:
  children:
    deploy_dev:
      hosts:
        dev-01.deploy.local:
          ansible_host: 192.168.101.202
```


A solution must be able to:

- ~~1. Configure servers and deploy applications~~
- ~~2. Run tasks on the controller, dev, and prod~~
- 3. Run tasks with host-specific variables
- 3a. Run tasks with host-specific **secrets**



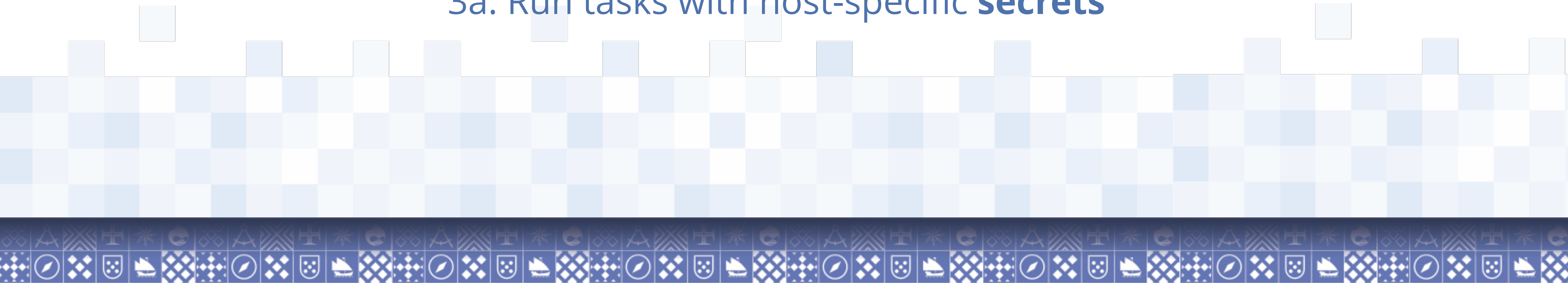
A solution must be able to:

~~1. Configure servers and deploy applications~~

~~2. Run tasks on the controller, dev, and prod~~

~~3. Run tasks with host specific variables~~

3a. Run tasks with host-specific **secrets**



Ansible Vault



Ansible Vault

encrypted *variables* or *files*



Ansible Vault

encrypted *variables* or *files*

--vault-password-file



Ansible Vault

encrypted *variables* or *files*

--vault-password-file

vault.yml →



Ansible Vault

```
$ANSIBLE_VAULT;1.1;AES256
30346634326166343330643433383738313666636566373230353837356533323562306232393963
3430316633373161313165623239386632623036666532370a313731613837303966336531623566
34643832313465626364643435633935313131306437386235373836303936393037373266396535
6463303132363939380a373232383935343862346339613534653633316661323136353233346239
66616239316538373435393039653630386236356135643866663932313165306432363164643030
32396262353662623339346132333333356335653633316633613161626337303465363761633836
33633165373434386532303739343330633434376639313133633138633035386261613630316536
33613537333832663534
```



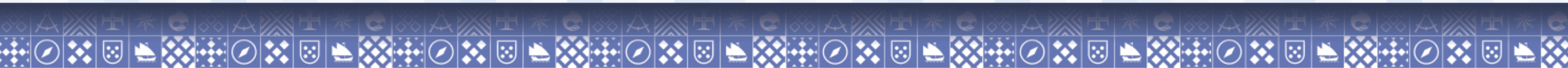
> `ansible-playbook deployment.yml`




```
> ansible-playbook deployment.yml  
  --vault-password_file=.deploy-vault
```



```
ctorgalson@bh-Macmini7-1: ~/deploy-everywhere/devops  
→ devops git:(master) x ansible-playbook deployment.yml -l deploy_dev -t deploy
```



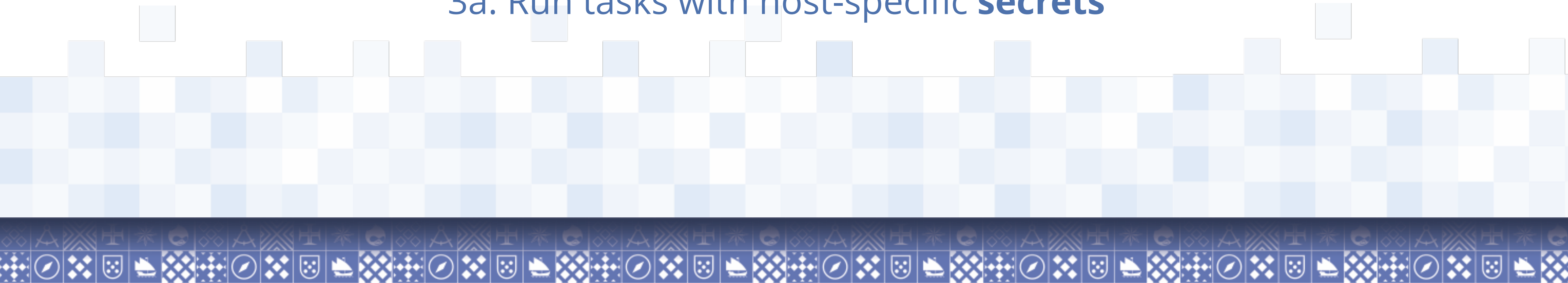
A solution must be able to:

~~1. Configure servers and deploy applications~~

~~2. Run tasks on the controller, dev, and prod~~

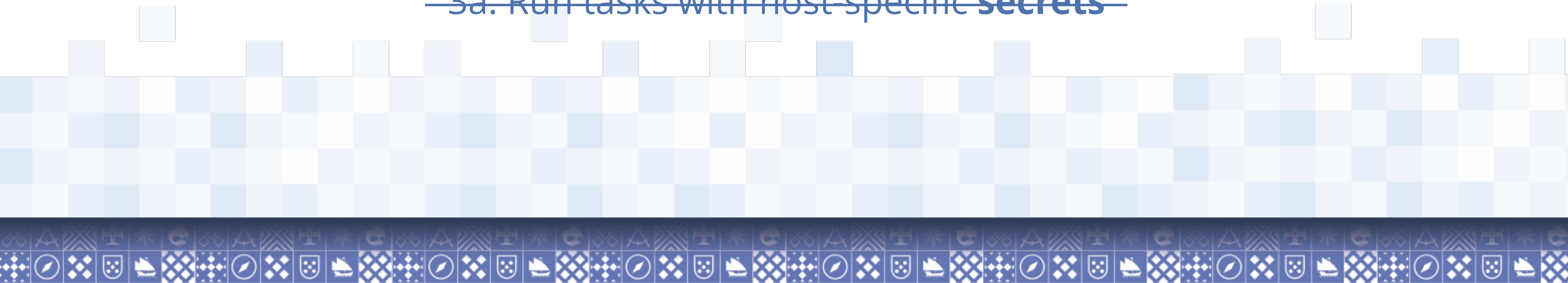
~~3. Run tasks with host specific variables~~

3a. Run tasks with host-specific **secrets**



A solution must be able to:

- ~~1. Configure servers and deploy applications~~
- ~~2. Run tasks on the controller, dev, and prod~~
- ~~3. Run tasks with host specific variables~~
- ~~3a. Run tasks with host specific **secrets**~~



Deployment



What tasks?



What tasks?

run build tasks



What tasks?

run build tasks

transfer codebase



What tasks?

run build tasks

transfer codebase

customize files



What tasks?

run build tasks

transfer codebase

customize files

execute commands



What tasks?

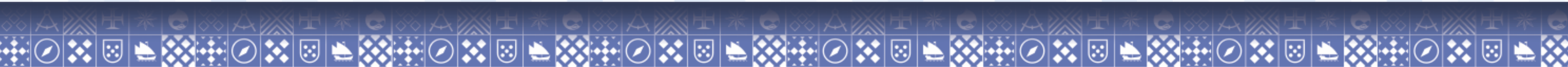
run build tasks

transfer codebase

customize files

execute commands

deploy-app.yml →



Run build tasks



Run build tasks

```
run npx gulp build
```



Run build tasks

```
19  
20 - name: Run Gulp build.  
21   command: "npx gulp build"  
22   args:  
23     chdir: "../site/docroot"  
24   delegate_to: localhost  
25
```



Transfer codebase



Transfer codebase

rsync files



Transfer codebase

```
25  
26 - name: Rsync site files into place.  
27   synchronize:  
28     src: "{{ playbook_dir | dirname }}/site/"  
29     dest: "{{ app_vhost_dir }}"  
30     private_key: "{{ ansible_ssh_private_key_file }}"  
31     when: "'prod' in app_hostname"  
32
```

Customize Files



Customize Files

Customize index.html



Customize Files

```
32  
33 - name: Template index.html file into place.  
34   template:  
35     src: "index.html.j2"  
36     dest: "{{ app_vhost_dir }}/{{ app_docroot_dir }}/index.html"  
37     owner: "{{ app_user }}"  
38     group: "{{ app_user }}"  
39
```

Execute Commands



Execute Commands

```
run npx gulp build (redux)
```



Execute Commands

```
19  
20 - name: Run Gulp build.  
21   command: "npx gulp build"  
22   args:  
23     chdir: "../site/docroot"  
24   delegate_to: localhost  
25
```

Extras



Git Hooks



Git Hooks

Run scripts on git actions



Git Hooks

Run scripts on git actions

Can be used to run tests



Git Hooks

Run scripts on git actions

Can be used to run tests

`--no-verify`



Git Hooks

Run scripts on git actions

Can be used to run tests

`--no-verify`

pre-push →



> `git push origin master`




```
ctorgalson@bh-Macmini7-1: ~/deploy-everywhere/devops  
→ devops git:(master) x git push origin master
```



> `git push -n origin master`



```
ctorgalson@bh-Macmini7-1: ~/deploy-everywhere/devops  
→ devops git:(master) x git push --no-verify origin master
```

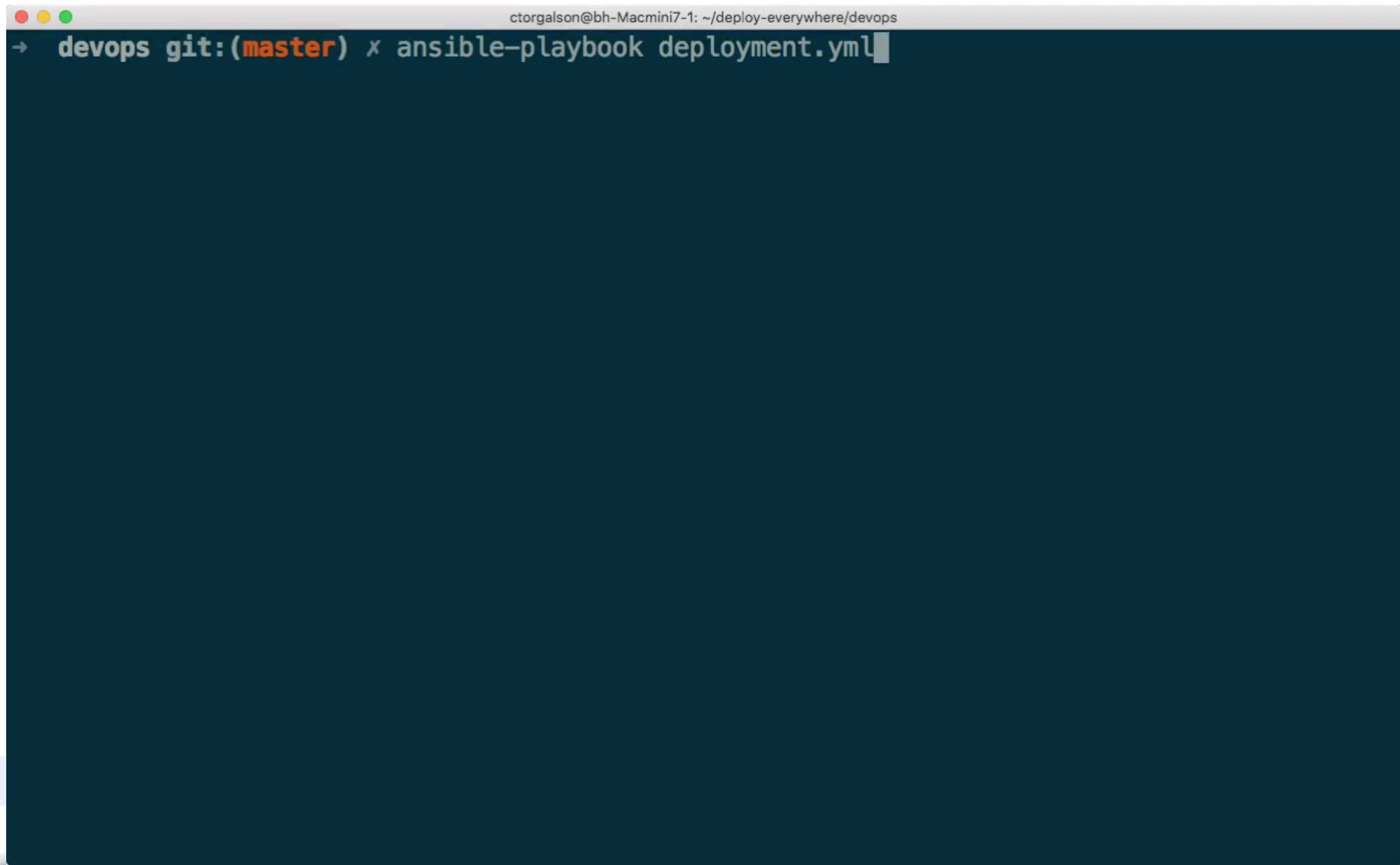


(Fake) Demo



> `ansible-playbook deployment.yml`





Deployed



